



**Pós-Graduação em Ciência da Computação**

**“AVALIAÇÃO DA UTILIZAÇÃO DE MELHORIA DE  
PROCESSO DE SOFTWARE BASEADA EM  
METODOLOGIAS ÁGEIS EM EMPRESAS CMMI”**

**Por**

***CÉLIO ANDRADE DE SANTANA JÚNIOR***

**Tese de Doutorado**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
www.cin.ufpe.br/~posgraduacao

RECIFE, Agosto/2012



UNIVERSIDADE FEDERAL DE PERNAMBUCO

CENTRO DE INFORMÁTICA

PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

CÉLIO ANDRADE DE SANTANA JÚNIOR

“AVALIAÇÃO DA UTILIZAÇÃO DE MELHORIA DE  
PROCESSO DE SOFTWARE BASEADA EM  
METODOLOGIAS ÁGEIS EM EMPRESAS CMMI”

ESTE TRABALHO FOI APRESENTADO À  
PÓS-GRADUAÇÃO EM CIÊNCIA DA  
COMPUTAÇÃO DO CENTRO DE  
INFORMÁTICA DA UNIVERSIDADE FEDERAL  
DE PERNAMBUCO COMO  
REQUISITO PARCIAL PARA OBTENÇÃO DO  
GRAU DE DOUTOR EM CIÊNCIA DA  
COMPUTAÇÃO.

ORIENTADOR: ALEXANDRE M. L. VASCONCELOS

RECIFE, AGOSTO/2012

Tese de Doutorado apresentada por Célio Andrade de Santana Júnior à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título *õ*AVALIAÇÃO DA UTILIZAÇÃO DE MELHORIA DE PROCESSO DE SOFTWARE BASEADA EM METODOLOGIAS ÁGEIS*ö* orientada pelo Prof. Alexandre Marcos Lins de Vasconcelos e aprovada pela Banca Examinadora formada pelos professores:

---

Prof. Sergio Castelo Branco Soares  
Centro de Informática / UFPE

---

Prof. Fernando José Castor de Lima Filho  
Centro de Informática / UFPE

---

Prof. Vinicius Cardoso Garcia  
Centro de Informática / UFPE

---

Prof. Alfredo Goldman vel Lejbman  
Departamento de Ciências da Computação / USP

---

Prof. Fabio Kon  
Departamento de Ciência da Computação/USP

Visto e permitida a impressão.  
Recife, 18 de junho de 2012.

---

**Prof. Nelson Souto Rosa**

Coordenador da Pós-Graduação em Ciência da Computação do  
Centro de Informática da Universidade Federal de Pernambuco.



*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

Catálogo na fonte  
Bibliotecária Joana D'Arc L. Salvador, CRB 4-572

Santana Júnior, Célio Andrade de.

Avaliação da utilização de melhoria de processo de software baseada em metodologias ágeis em empresas CMMI+/ Célio Andrade de Santana Júnior. . Recife: O Autor, 2012.

xii, 213 p.: fig., tab.

Orientador: Alexandre M. L. de Vasconcelos.

Tese (Doutorado) - Universidade Federal de Pernambuco. CIN. Ciência da Computação, 2012.

Inclui bibliografia, apêndice e anexo.

1. Engenharia de software. 2. Modelos de capacitação e maturidade (software).  
3. Desenvolvimento ágil de Software. I. Vasconcelos, Alexandre M. L. de (orientador). II. Título.

005.1 (22. ed.) MEI 2012-098

## Resumo

Em 2009 a Associação para Promoção da Excelência do Software Brasileiro (SOFTEX<sup>1</sup>) publicou uma pesquisa indicando que 96% das empresas de software brasileiras eram pequenas e possuíam até 20 pessoas. Ainda que estas empresas empregassem 55,3% de toda mão de obra da área, elas respondiam por apenas 19% do faturamento do setor e por menos de 1% das exportações de software. A conclusão do diagnóstico aponta que essas empresas precisavam passar por reestruturações em seus processos para aumentar sua competitividade. Algumas dessas empresas buscaram essa melhoria partindo para programas de certificação de qualidade tais como o CMMI e MPS.BR. Este trabalho propõe avaliar a utilização de metodologias ágeis como direcionador de programas de melhoria de processo, com o objetivo de atingir níveis de maturidade em avaliações CMMI nível dois e MPS.BR nível F. Uma contribuição deste trabalho é gerar evidências empíricas sobre a adoção de métodos ágeis como política de melhoria de processo em empresas de vida real no Brasil. Assim, esta pesquisa foi realizada para responder as seguintes questões de pesquisa: QP1 ó É possível executar programas de melhoria de processo de software baseados em métodos ágeis com o objetivo de certificação CMMI nível dois e MPS.BR nível F? QP2 ó Quais os benefícios observados por estas organizações ao se adotar a melhoria de processo baseada em métodos ágeis? E em longo prazo? QP3 ó Como a melhoria de processo ágil está relacionada à melhoria de processo prescritiva adotada pelo CMMI e MPS.BR. Esta tese resume quatro anos de pesquisa em oito pequenas e médias empresas Brasileiras. Os métodos utilizados para a pesquisa foram: (i) estudo de caso baseado em avaliações de 3º pessoa registradas em planilhas para responder a QP1; (ii) *surveys* e (iii) teoria fundamentada em dados foi utilizada para responder as questões QP2 e QP3.

**Palavras Chaves:** *Scrum; Capability Maturity Model Integration (CMMI); Desenvolvimento Ágil de Software; Melhoria de Processo de Software.*

---

<sup>1</sup> www.softex.br

## Abstract

In 2009 the Brazilian Association for Promoting Software Excellence (SOFTEX<sup>2</sup>) published a survey indicating that 96% of Brazilian software companies were small with up to 20 people. Although these firms employ 55.3% of the entire workforce of the area, they accounted for only 19% of all sales of the sector and for less than 1% of software exports. The conclusion of the diagnosis points out that these companies had to pass on restructuring processes to increase their competitiveness. Some of these companies sought this improvement by starting programs for quality certifications such as CMMI and MPS.BR. This work aims to evaluate the use of agile methods as a director of process improvement programs with the goal of reaching maturity levels in CMMI level two and MPS.BR level F assessments. One contribution of this work is to generate empirical evidence on the adoption of agile methods, not as tools but as political process improvement in real-life companies in Brazil. Knowing this, this research was motivated to answer the following research questions: RQ1 - Is it possible to run software process improvement programs based on agile methodologies to reach CMMI staged level two?; RQ2 - What are the benefits observed by these organizations when adopting this way of process improvement, including the long term?; RQ3 - How agile process improvement is related to process improvement (*prescriptive*) adopted by the CMMI and MPS.BR? This thesis summarizes four years of research on eight small and medium-sized Brazilian companies. Our methodologies comprises: (i) a case study based on assessments of third person registered in spreadsheets QP1; (ii) surveys and (iii) grounded theory was the data used to answer questions QP2 and QP3.

**Keywords:** Scrum, Capability Maturity Model Integration, Agile Software Process Improvement.

---

<sup>2</sup> [www.softex.br](http://www.softex.br)

## Listade Imagens

Figura 1 – Evolução do Faturamento das Empresas Certificadas MPS.BR.....	23
Figura 2 - Evolução do Custo dos Projetos das Empresas Certificadas MPS.BR.....	23
Figura 3 - Evolução do Prazo Médio dos Projetos das Empresas Certificadas MPS.BR	24
Figura 4 - Evolução da Produtividade das Empresas Certificadas MPS.BR.....	24
Figura 5 - Evolução da Qualidade dos Produtos das Empresas Certificadas MPS.BR..	24
Figura 6 - Evolução da ROI das Empresas Certificadas MPS.BR.....	25
Figura 7 – Quando organizações podem se definir ágeis [Ambler, 2009] .....	49
Figura 8 – Por que projetos ágeis falham [Versionone, 2011] .....	51
Figura 9 – Passos para Realização de um Estudo de Caso [Brereton et al, 2008].....	69
Figura 10 - Taxa de Sucesso em Programas de MPS. ....	85
Figura 11 – Comparativo do faturamento das empresas iMPS e as empresas pesquisadas.....	88
Figura 12 – Comparativo do custo médio por projetos das empresas iMPS e as empresas pesquisadas.....	88
Figura 13 – Comparativo do prazo médio por projetos das empresas iMPS e as empresas pesquisadas.....	89
Figura 14 – Comparativo da produtividade das empresas iMPS e as empresas pesquisadas.....	89
Figura 15 – Comparativo da qualidade das empresas iMPS e as empresas pesquisadas.....	90
Figura 16 – Comparativo do faturamento das empresas iMPS e as empresas pesquisadas em porcentagem.....	90
Figura 17 – Comparativo do custo médio das empresas iMPS e as empresas pesquisadas em porcentagem.....	91
Figura 18 – Comparativo do prazo médio das empresas iMPS e as empresas pesquisadas em porcentagem.....	91
Figura 19 – Comparativo da produtividade das empresas iMPS e as empresas pesquisadas em porcentagem.....	92

Figura 20 – Comparativo da qualidade das empresas iMPS e as empresas pesquisadas em porcentagem.....	92
Figura 21 – Evolução dos Indicadores das Organizações após as avaliações dos modelos de Maturidade.....	94
Figura 22 – Resultados da empresa que sofreu a maior erosão de processo para o CMMI .....	95
Figura 23 – Resultados da empresa que sofreu a menor erosão de processo para o CMMI .....	96
Figura 24 – Resultados da empresa que sofreu a maior erosão de processo para o MPS.BR .....	96
Figura 25 – Resultados da empresa que sofreu a menor erosão de processo para o MPS.BR .....	97
Figura 26 – Erosão das Práticas Genéricas do CMMI .....	97
Figura 27 – Erosão dos Atributos de Processo do MPS.BR .....	98
Figura 28 – Maior Erosão de Processo apontada pelo Nokia Test.....	98
Figura 29 – Menor Erosão de Processo apontada pelo Nokia Test.....	99
Figura 30 – Nuvem de palavras dos questionários recebidos pelas empresas sem considerar a palavra mudança.....	100
Figura 31 – Nuvem de palavras dos questionários recebidos pelas empresas sem considerar a palavra mudança.....	101
Figura 32 – Codificação coaxial dos códigos apresentados nos questionários .....	102
Figura 33 -Estruturas Hierárquicas x Estruturas Adocráticas [Mintzberg e McHugh, 1985].....	109

## Lista de Tabelas

Tabela 1 – Indicadores Globais do iMPS [Travassos e Kalinowsky, 2008 a 2011].....	19
Tabela 2 – Indicadores Globais do iMPS [Travassos e Kalinowsky, 2008 a 2011].....	22
Tabela 3 – Relacionamento entre os trabalhos, principais resultados e as questões de pesquisa.....	36
Tabela 4- Relacionamento entre os trabalhos publicados e as empresas.....	36
Tabela 5- Estrutura do Documento .....	38
Tabela 6. Motivos para a adoção de programas de MPS baseados em modelos de Maturidade [Staples e Niazi, 2007]. .....	43
Tabela 7. Motivos para a não adoção de programas de MPS baseados em modelos de Maturidade [Staples et al., 2007]. .....	44
Tabela 8. Fatores de Sucesso na Implementação do MPS.BR e CMMI em empresas Brasileiras [Rocha et al., 2005].....	45
Tabela 9. Fatores de Insucesso na Implementação do MPS.BR e CMMI em empresas Brasileiras [Rocha et al., 2005].....	45
Tabela 10. Diferenças entre as abordagens de MPS Tradicional e Ágil. ....	56
Tabela 11 - Quadro Metodológico.....	68
Tabela 12 – Comparativo dos Indicadores Gerais das Empresas pesquisadas x iMPS	86
Tabela 13 – Artigos Publicados .....	162

## Lista de Siglas e Acrônimos

- ARC ó Requisitos de Avaliação do CMMI;
- CMM - *Capability Maturity Model*;
- CMMI ó *Capability Maturity Model Integration*;
- ISO - *International Organization for Standardization*;
- MPS ó Melhoria de Processo de Software;
- MPS.BR ó Programa para a Melhoria de Processo do Software Brasileiro;
- OTAN ó Organização do Tratado do Atlântico Norte;
- PME ó Pequenas e Médias Empresas;
- PPQA ó Comitê de Garantia de Qualidade do Processo e do Produto;
- QP ó Questão de Pesquisa;
- ROI ó Retorno de Investimento;
- SAC ó Sistemas Adaptativo Complexo;
- SEI ó *Software Engineering Institute*;
- SOFTEX - Associação para Promoção da Excelência do Software Brasileiro.
- TI ó Tecnologia da Informação;
- XP ó Extreme Programming.

## Sumário

<b>Resumo</b> .....	IV
<b>Abstract</b> .....	V
<b>Lista de Imagens</b> .....	VI
<b>Lista de Tabelas</b> .....	VIII
<b>Lista de Siglas e Acrônimos</b> .....	IX
<b>Sumário</b> .....	X
<b>1. Introdução</b> .....	13
1.1. Lacunas de Pesquisa.....	16
1.2. Motivação e Questões de Pesquisa .....	19
1.3. Objetivos.....	29
1.4. Contexto de Pesquisa.....	30
1.5. Contribuição e Resultados Esperados .....	33
1.6. Estrutura da Tese .....	38
<b>2. Referencial Teórico</b> .....	39
2.1 Melhoria de Processo de Software.....	39
2.2 Melhoria de Processo De software baseada em modelos de maturidade ..	41
2.3 Mentalidade Ágil .....	46
2.4 Melhoria de Processo de Software Ágil.....	51
2.5 Erosão de Processo .....	56
2.6 Trabalhos Relacionados .....	58
2.7 Resumo do capítulo .....	64
<b>3. Metodologia de Pesquisa</b> .....	66
3.1. Abordagem de pesquisa .....	66
3.2. Construção da Pesquisa .....	68
3.3. sumário do capítulo .....	83

<b>4. Análise dos Resultados.....</b>	<b>84</b>
4.1. QP1 - É possível executar programas de melhoria de processo de software baseados em metodologias ágeis com o objetivo de certificação CMMI nível dois e MPS.BR nível F? .....	84
4.2. QP2 – Quais os benefícios observados por estas organizações ao se adotar a melhoria de processo Baseada em métodos ágeis? E em longo prazo? .....	86
4.3. QP3 – Como a melhoria de processo ágil “Indutiva” está relacionada à melhoria de processo “Prescritiva” adotada pelo CMMI e MPS.BR? .....	99
4.4. sumário do capítulo .....	119
<b>5. Conclusão.....</b>	<b>121</b>
5.1. Descobertas.....	121
5.1.1. <i>Empresas de Software podem obter níveis de maturidade em modelos de qualidade utilizando abordagens para melhoria de processo baseado em métodos ágeis.</i> 121	
5.1.2. <i>Quais os Benefícios Observados Por Estas Organizações ao se adotar esta forma de Melhoria de Processo? E em longo prazo? .....</i>	121
5.1.3. <i>Como a melhoria de processo ágil “indutiva” está relacionada À melhoria de processo “prescritiva” adotada pelo CMMI e MPS.BR? .....</i>	122
5.1.4. <i>Negócio é a base para a melhoria de processo ágil? .....</i>	123
5.1.5. <i>As Organizações Ágeis+CMMI ou Ágeis+MPS.BR não são híbridas? .....</i>	124
5.1.6. <i>Modelos de Qualidade e Abordagens Ágeis apresentam choque de cultura, principalmente se adotados em conjunto. ....</i>	125
5.1.7. <i>Os times nestas organizações não são auto-gerenciados; .....</i>	125
5.1.8. <i>O Aprendizado NO MPS ágil necessita de mecanismos diferentes daqueles providos no MPS Tradicional. ....</i>	125
5.1.9. <i>A Cultura Organizacional é Percebida no CMMI e MPS.BR A partir DE práticas Genéricas e Atributos dos Processos Respectivamente. ....</i>	126
5.1.10. <i>A Melhoria Adaptativa de Processo Acontece no Nível de Time e não de Organização; .....</i>	126
5.1.11. <i>A Erosão do processo se dá por que muitos dos times não estão alinhados com a cultura da organização ou existe a falta da visibilidade do valor de negócio do processo para o time; .....</i>	126

5.1.12. Coarjicaao Coaxial Dos Fatores Envolvidos Na Junção de Metodologias Ágeis e Modelos de Qualidade. ....	127
5.2. Discussão dos Resultados.....	128
5.3. Limitações e Ameaças à Validade.....	131
5.4. Implicações Práticas e Teóricas.....	133
5.5. Trabalhos Futuros.....	134
<b>Referências.....</b>	<b>136</b>
<b>Apêndice A ó Email de Apresentação.....</b>	<b>148</b>
<b>Apêndice B ó 1º Levantamento de Informações.....</b>	<b>150</b>
<b>Apêndice C ó 2º Levantamento de Informações.....</b>	<b>151</b>
<b>Apêndice D ó 3º Levantamento de Informações.....</b>	<b>153</b>
<b>Apêndice E ó Empresas que não seguiram com o CMML.....</b>	<b>160</b>
<b>Apêndice F ó Artigos Publicados.....</b>	<b>162</b>
<b>Anexo A ó Nokia Test.....</b>	<b>212</b>

## 1. Introdução

O crescente papel da tecnologia da informação (TI) na economia moderna criou exigências maiores relativas à qualidade do produto de software produzido hoje em dia. Neste contexto, a Melhoria do Processo de Software (MPS) se tornou a principal abordagem para a melhoria da qualidade do software e satisfação dos clientes [Mathiassen, Ngwenyama e Aaen, 2005].

O ambiente no qual o software é projetado e criado também está em constante mudança. Os sistemas de software estão se tornando maiores e mais complexos e o ritmo de mudanças nos requisitos é crescente. Mesmo depois de 44 anos, a indústria de software atual ainda apresenta alguns dos desafios lançados na conferência da Organização do Tratado do Atlântico Norte (OTAN) quando o termo Engenharia de Software foi criado [Naur e Randell, 1968; Moe, 2011].

A existência dessas dificuldades ainda nos dias de hoje é sugerida pelo relatório *Chaos Manifesto* publicado em 2012 [Standish Group, 2012]. Este relatório é publicado desde 1994 ainda quando possuía o nome de *Chaos Report* e apresenta dados referentes a mais de 10 mil projetos. Entre estes dados estão a taxa de sucesso, tipo de ciclo de vida usado, linguagens utilizadas e etc.. A partir destes dados alguns indicadores podem ser inferidos e alguns de muito interesse para esta pesquisa como por exemplo: este relatório mostra que 29% dos projetos que utilizam o modelo cascata não são sequer colocados em produção; seja por que foram abortados ou porque o produto criado não atendia as necessidades dos clientes. Ainda considerando os projetos cascata, apenas 14% foram classificados como bem sucedidos, ou seja, eram concluídos com o escopo completo e com estouro de prazo e/ou orçamento em até 50%. [Standish Group, 2012].

Para entendermos melhor a motivação para o surgimento dos métodos ágeis é necessário entender o contexto tradicional do desenvolvimento de software no início da primeira década de 2000 quando esses surgiram. Naquela época, alguns dos pesquisadores da Engenharia de Software enxergavam o

ambiente de desenvolvimento de software tradicional como caracterizado pela abordagem da linha de produção. Essa abordagem se caracterizava pela tentativa de adoção de um sistema padronizado a partir de um processo controlável e previsível de software [Dybå 2000; Boehm, 2002].

Essas abordagens tradicionais envolviam planejamento extenso, especificação dos processos e reutilização rigorosa para tornar o desenvolvimento de software uma atividade eficiente e previsível [Dybå 2000; Boehm, 2002]. Esta forma de desenvolvimento de software também é conhecida como abordagem orientada a planejamento e é geralmente guiada por um modelo de ciclo de vida, com atenção à qualidade dos artefatos de software e à previsibilidade de seus processos [Lycett et al. 2003].

Esta abordagem da engenharia favorece os processos explicitamente definidos, que podem ser padronizados. O aumento da complexidade dos projetos de software é dado pela dependência dessa previsão de desenvolvimento, que por algumas vezes pode impactar adversamente quando houver mudanças no sistema [Boehm e Turner, 2003].

A comunidade de engenharia de software gradualmente abandonou a qualidade com ênfase no produto final por aquela com ênfase no processo. Assim, métodos corretivos baseados em inspeções no final da linha de produção foram substituídos por processos e técnicas de controle de qualidade durante todo o ciclo de vida do desenvolvimento e é nessa visão que MPS ganha importância na qualidade de software [Georgiadou et al., 2003].

Geralmente, considera-se que um processo bem documentado e repetível é essencial para o desenvolvimento de produtos de software de alta qualidade. Em se tratando de modelos de qualidade, existem evidências apontando que o uso de padrões e modelos de avaliação de processo resulta em melhores resultados não só relativos ao custo e cronograma dos projetos, mas também à qualidade do produto final de software [Kitchenham e Pfleeger, 1996].

A engenharia de software tradicional tem preferido a adoção da abordagem prescritiva de melhoria de processo de software. Esta abordagem se preocupa em adotar um processo para a organização a partir de um guia de boas

práticas. A suposição geral é que quanto mais o processo de uma organização estiver adequado às estipulações do modelo, maior será a sua eficácia em alguns critérios tais como: produtividade, qualidade do produto e previsibilidade [Briand et al., 1995].

Na abordagem prescritiva todas as atividades devem ser executadas de forma estruturada e sistemática para que outras atividades em andamento não sejam perturbadas. Uma vez que a iniciativa da MPS seja realizada corretamente, esperam-se resultados positivos nos indicadores organizacionais e na margem de lucro da empresa. Aspectos como estrutura organizacional, guias de recomendação, métodos para a avaliação da melhoria e um bom planejamento, levam a resultados bem sucedidos na execução de iniciativas de melhoria de processos [Allison e Merali, 2007; Butt, 2007].

A evolução dos modelos de negócio envolvidas no desenvolvimento de software forçou os programadores a considerar os pontos de vista de uma grande variedade de partes interessadas, muitos dos quais têm visões conflitantes sobre a conveniência dos recursos do software e suas funcionalidades. Esse ambiente turbulento exige que os gestores coordenem e ajustem continuamente as prioridades de diversas necessidades de mudança [Nerur e Balijepally 2007].

No início da primeira década de 2000, surgiram os métodos ágeis que apresentavam princípios e técnicas divergentes da visão tradicional do desenvolvimento de software. Em contraste com as abordagens orientadas a planejamento, os métodos ágeis assumiam que era desafiador desenvolver software em um mundo imprevisível, em que as pessoas e a criatividade eram mais importantes do que os processos. Neste conceito, a busca pela otimização estava sendo substituída pela flexibilidade [Dybå 2000; Nerur, Mahapatra e Mangalaraj 2005, Nerur e Balijepally 2007].

Erickson et al. (2005) definiram agilidade como um meio para abandonar parte do peso associado ao desenvolvimento tradicional de software, para promover resposta rápida a mudanças em ambientes dinâmicos e com requisitos voláteis. Métodos Ágeis surgiram a partir de descobertas comuns de seus praticantes que abandonaram a visão tradicional, centrada em processos e

documentação pesada, em prol de abordagens centradas em pessoas e mais leves em termos de documentação [Fowler, 2002; Highsmith, 2002a; Boehm e Turner, 2003].

Métodos ágeis apresentam processos técnicos e gerenciais que se adaptam continuamente a mudanças advindas da experiência adquirida durante o desenvolvimento e principalmente às mudanças no ambiente de desenvolvimento [Dan Turk e Rumpe, 2002]. Evidentemente, essa diferença de mentalidade exigia também novas formas para a realização de MPS nestes ambientes. Os ciclos curtos de desenvolvimento possibilitam uma forma acelerada de aprendizado contínuo e a melhoria dos processos [Salo e Abrahamsson, 2006].

Moe (2011) afirma que existem diferenças entre o MPS tradicional e o MPS ágil, pois o desenvolvimento ágil requer novos mecanismos que se adaptem ao contexto de mudanças. O MPS tradicional prefere uma abordagem prescritiva com ênfase na padronização e aderência a um modelo, como é o caso do *Capability Maturity Model Integration* (CMMI) [Chrissis et al., 2011] e do Modelo de Melhoria de Processo do Software Brasileiro (MPS.BR) [MPS.BR, 2011a]. Já o desenvolvimento ágil prefere uma abordagem diferente para a MPS, que consiste em melhorar o processo de aprendizado e comportamento dos times, de modo que seja possível valorizar a experiência pessoal de cada membro. [Mathiassen, Ngwenyama & Aaen 2005; Salo 2007].

### 1.1. LACUNAS DE PESQUISA

As principais lacunas de pesquisa na área de Melhoria de Processo de Software em Métodos Ágeis são levantadas nos seguintes estudos acadêmicos:

**L1** - Um estudo realizado por Qumer e Henderson-Sellers (2008) identificou a dificuldade de adaptar métodos ágeis dentro de organizações. Segundo eles, esta dificuldade era refletida nos insucessos relatados pela indústria na adoção de métodos ágeis. Por isso, eles sugeriram a criação de um arcabouço que possa ser usado pelas empresas, para criar, modificar e adaptar

situações específicas envolvendo métodos ágeis. Este arcabouço deveria incluir modelos de medição de agilidade, bem como, guias para a adoção de métodos ágeis e um modelo de melhoria da agilidade dentro das organizações. Assim, a lacuna de pesquisa identificada a partir dessa pesquisa é a falta de métodos para a adaptação de processos ágeis em organizações de software.

**L2** - Moe (2011) alega que a melhoria de processo deve ocorrer em todos os níveis da organização, para assim permitir o completo autogerenciamento em ambientes dirigidos a mudanças (ágeis). Entretanto, nenhuma das organizações pesquisadas por ele conseguiu estabelecer um estado de completo autogerenciamento. Então, os trabalhos futuros deveriam focar em determinar que características de MPS são observadas nesse tipo de ambiente. Moe ainda relata que é necessário determinar como envolver toda a organização na adoção do desenvolvimento dirigido a mudanças e observar os resultados no MPS neste tipo de organização que apresenta um ambiente dinâmico e turbulento.

**L3** ó Existe a necessidade de se pesquisar a área emergente de desenvolvimento global de software e como ela está se relacionando com os métodos ágeis, principalmente no tocante à melhoria de processos e ao aprendizado organizacional [Smite, Moe e Ågerfalk, 2010]. Nesta área são pesquisados temas como desenvolvimento distribuído de software, entrega contínua e escalabilidade de projetos. Segundo Smite et al. (2010) não há evidências empíricas sobre como uma melhoria global pode ser conduzida de forma ágil.

**L4** ó Salo (2007) levantou algumas lacunas de pesquisa relacionadas à MPS em ambientes ágeis, tais como a necessidade de identificar os mecanismos de aprendizado e a gestão de conhecimento em ambientes em que a melhoria do comportamento do time é o núcleo do MPS. Então, a lacuna de pesquisa em questão se refere a como a experiência e conhecimento obtido pelo time pode ser aproveitado pela organização.

**L5** ó Salo (2007) identifica outra lacuna de pesquisa que pode ser elencada em dois pontos: (i) como deve ocorrer a integração entre a melhoria contínua promovida pelo MPS ágil e o MPS tradicional prescritivo e (ii) como

ocorre esse MPS o conjunto dentro destas organizações.

**L6** ó Salo (2007) ainda levanta que é necessário pesquisar como ocorre o MPS ágil dentro de organizações que possuem ou tentam alcançar níveis de maturidade em modelos de qualidade como o CMMI e MPS.BR. Esta mesma lacuna é identificada por Dyba e Dyngsoir (2008) em uma revisão sistemática. Segundo os autores ainda se deve abordar a relação entre os padrões de processo e o desenvolvimento de software ágil.

**L7** ó Dingsoyr et al. (2012) afirma que ainda seriam necessários estudos explicativos para testar as teorias e fornecer uma compreensão do processo de aprendizagem que ocorre dentro das equipes ágeis ao se conduzir reflexões iterativas para a melhoria de processo de software. Esta lacuna complementa a Lacuna L4 que esta está preocupada em transformar o conhecimento obtido pela equipe em conhecimento organizacional. Esta Lacuna L7 está relacionada com compreender o processo (intra-time) de transformar o conhecimento em melhoria de comportamento e a evolução dessa melhoria a partir das iterações.

**L8** - Considerando as lacunas que envolvem diretamente a relação dos métodos ágeis e o CMMI e MPS.BR, Pikkarainen (2008) e Santana et al. (2009a) afirmam que são necessárias pesquisas tratando dos níveis de maturidade mais altos do CMMI (nível 3 em diante).

**L9** ó Pikkarainen (2008) ainda afirma que a pesquisa sobre a adoção de práticas ágeis em ambientes CMMI precisa ser enriquecida com evidências empíricas por meio de pesquisas em larga escala. Isto poderia ser usado para determinar os níveis dos métodos ágeis e de assimilação das práticas ágeis dentro das organizações de software.

**L10** - Santana et. al. (2009a) levanta a necessidade de se fazer mais pesquisas sobre como as práticas genéricas do CMMI poderiam apoiar a adoção de métodos ágeis, uma vez que são as práticas genéricas do CMMI que determinam o nível de maturidade de uma organização.

Abaixo é apresentada a Tabela 1 a seguir com um sumário das lacunas de pesquisa.

Tabela 1—Indicadores Globais do iMPS [Travassos e Kalinowsky, 2008 a 2011].

Lacuna	Referência	Assunto
<b>L1</b>	[Qumer e Henderson-Sellers, 2008]	Criação de um Framework para auxiliar a implantação de abordagens ágeis.
<b>L2</b>	[Moe, 2011]	Estudo do MPS ágil em empresas que apresentam estado de completo autogerenciamento.
<b>L3</b>	[Smite, Moe e Ågerfalk, 2010]	Estudo do MPS utilizando métodos ágeis em ambientes globais.
<b>L4</b>	[Salo, 2007]	Como o aprendizado intra-times baseado na experiência do indivíduo pode ser levado ao nível organizacional?
<b>L5</b>	[Salo, 2007]	Como a integração entre MPS ágil e MPS tradicional nas organizações?
<b>L6</b>	[Salo, 2007]	Como ocorre a integração proposta na L5 em ambientes CMMI?
<b>L7</b>	[Dingsoyr et al, 2012]	Como reflexões iterativas levam o time a melhoria continua?
<b>L8</b>	[Pikkarainen, 2008], [Santana et al, 2009a]	Como ocorre o uso de métodos ágeis na melhoria de processo dentro de organizações com níveis de maturidade mais altos do CMMI?
<b>L9</b>	[Pikkarainen, 2009]	Necessidade de mais dados empíricos sobre a adoção de métodos ágeis em CMMI.
<b>L10</b>	[Santana et al, 2009a]	Investigação do papel das práticas genericas do CMMI em ambientes CMMI + Ágil.

## 1.2. MOTIVAÇÃO E QUESTÕES DE PESQUISA

Um diagnóstico sobre a indústria de software brasileira foi publicado em 2009 pela Associação para Promoção da Excelência do Software Brasileiro (SOFTEX) [SOFTEX, 2009]. Em 2009, havia no Brasil cerca de 32.000 empresas de desenvolvimento de software onde 96 % delas possuíam até 20 empregados. Entretanto, apenas 19% do faturamento bruto do setor está concentrado neste nicho de empresas. Este número cai para menos de 1% quando consideramos apenas as exportações. Este resultado é pior do que aqueles observados em outros setores como, por exemplo, na maior atividade econômica do Brasil, o agronegócio. No agronegócio brasileiro, 84,4% dos produtores pertencem à agricultura familiar e a participação destes produtores no mercado é de 38%. A

média de pessoas trabalhando em cada unidade agrícola neste tipo de negócio era de três pessoas por latifúndio [IBGE, 2009].

A agricultura familiar, mesmo muitas vezes se tratando de um processo industrial artesanal, com uma média de três pessoas por empresa e com uma escala de negócios entre 15 a 20 vezes maior que a indústria de software no Brasil, apresenta números apontando que os pequenos produtores possuem o dobro de participação, 38% *versus* 19%, na renda bruta do setor produtivo. Este número no setor de TI também está abaixo da média nacional, segundo o Serviço Brasileiro de Apoio a Pequena Empresa, SEBRAE (2011), em 2010 no Brasil 99% das empresas existentes eram pequenas e microempresas, e estas eram responsáveis por um terço de todo produto interno bruto do país.

Esta situação evidencia que a participação das pequenas empresas de tecnologia brasileira é pequena, e é percebida a necessidade do aumento de competitividade destas empresas. O Governo Brasileiro já havia observado este problema, tanto que através do SOFTEX o governo federal lançou o Programa Brasileiro de Melhoria de Processo de Software (MPS.BR), que incluía naquele momento dois objetivos [Weber et al., 2005]: (i) a criação de um modelo de melhoria de processo compatível com o CMMI [Chrissis et al, 2011], ISO 15504 [Van Loon, 2007] e ISO 12207 [ISO, 1995]; e (ii) Implementação e avaliação de programas MPS no Brasil a um custo acessível e com foco em pequenas e médias empresas (PME).

O governo brasileiro reconheceu que, com recursos limitados, as pequenas empresas geralmente precisam de ajuda externa para planejar e implementar a melhoria de processos. O apoio promovido pelo programa MPS.BR, desde o seu início, resultou em 363 avaliações MPS.BR bem sucedidas, e a procura por tais avaliações tem crescido. Em 2005 foram apenas cinco avaliações enquanto em 2011 foram 71. Isso mostra o aumento da procura das empresas brasileiras por este modelo específico [SOFTEX, 2012]. Entretanto, em números absolutos, apenas 1% foram beneficiadas por este programa.

Muitas das empresas estão aproveitando o programa MPS.BR para

meiorar a própria competitividade, com o objetivo de exportar software [Antonioni, 2010]. Entretanto, as empresas que pretendem exportar seus produtos buscam a certificação internacional CMMI, que se instituiu como um padrão em todo o mundo. Segundo o *Software Engineering Institute* (SEI) (SEI, 2012), houve 105 avaliações CMMI bem sucedidas em empresas brasileiras desde 2009. Cerca de 40% dessas empresas possuem as duas certificações (MPS.BR e CMMI). O que é observado é que as empresas que têm como meta o CMMI estão adotando a estratégia de iniciar com o MPS.BR e, a partir de então adotar para o CMMI.

Segundo o *Software Engineering Institute* (SEI) (SEI, 2012), houve 105 avaliações CMMI em empresas brasileiras de 2009 até hoje, e cerca de 40% dessas empresas também possuem o certificado MPS.BR. O que é observado é que as empresas que desejam o CMMI estão adotando a estratégia de começar pelo MPS.BR, e a partir de então alcançar o CMMI. Um total de 240 empresas certificaram MPS.BR de 2009 até os dias atuais.

Um fato a ser observado é que a procura pela certificação MPS.BR na indústria Brasileira se mostra aquém da expectativa do próprio SOFTEX. Eram esperadas 377 avaliações MPS.BR até Dezembro de 2010 [Webber et al., 2008], entretanto, em meados de 2012 é que se chegou a 363 avaliações. Esta baixa adesão pode ser resultado da percepção das empresas que é necessário primeiro investir em capacitação para depois investir em certificação. Este ponto é evidenciado por Rocha et al. (2005), que aponta que o principal fator de insucesso em programas de MPS.BR é devido à falta de competência das equipes.

Cabe aqui a observação de que embora a atitude do governo Brasileiro seja louvável, talvez seja necessário um investimento inicial mais voltado para uma política de capacitação das empresas brasileiras de TI, ao invés de adotar uma política baseada na melhoria de processo a partir de um modelo de certificação. Esta atual política reduz o alcance da iniciativa, já que apenas aproximadamente 1% das empresas brasileiras foram certificadas e, ao mesmo tempo não garante uma melhoria efetiva na indústria de TI nacional.

Esta Tabela de resultados inquestionáveis é apresentada no diagnóstico sobre a adoção do MPS.BR na indústria brasileira, que vem sendo publicado desde 2008 por Travassos e Kalinowsky (2008 a 2011). Esta pesquisa anual apresenta alguns indicadores da utilização do MPS.BR nas empresas e é chamada de iMPS. A Tabela 2 apresenta alguns dos indicadores globais da pesquisa iMPS ao longo dos anos. Os indicadores apresentados são:

- **Empresas:** Quantidade de empresas que responderam o questionário;
- **Tempo Médio:** Tempo médio para implementação do MPS.BR;
- **Custo Médio:** Custo da implementação relativo ao faturamento bruto da empresa;
- **Satisfação:** Satisfação das empresas com o MPS.BR;
- **Funcionários:** Número médio de funcionários das empresas respondentes.

**Tabela 2—Indicadores Globais do iMPS [Travassos e Kalinowsky, 2008 a 2011].**

Ano	Empresas	Tempo Médio	Custo Total	Satisfação
2008	123	12 Meses	5%	94,4%
2009	135	16,5 Meses	2,8%	98,6%
2010	156	12 Meses	3,4%	92,9%
2011	133	12 Meses	10%	97,0%

A partir do ano de 2009, o iMPS passou a apresentar alguns indicadores que apresentam o desempenho das organizações após a certificação MPS.BR. Nem todas as empresas responderam os dados referentes a evolução e assim a evolução desses indicadores, que serão mostrados nas imagens referentes a Figuras 1 à 5, serão apresentados em números absolutos em quantidade menor do que o total de respondentes no ano devido. Os indicadores avaliados foram:

- **Faturamento:** Aumento ou diminuição do faturamento bruto das organizações após a adoção do MPS.BR;
- **Custo Médio:** Aumento ou diminuição do custo médio por projeto das organizações após a adoção do MPS.BR;

- **Prazo Médio:** Aumento ou diminuição do tempo médio despendido por projeto das organizações após a adoção do MPS.BR;
- **Produtividade:** Aumento ou diminuição da produtividade das organizações após a adoção do MPS.BR;
- **Qualidade:** Aumento ou diminuição da qualidade dos produtos após a adoção do MPS.BR;

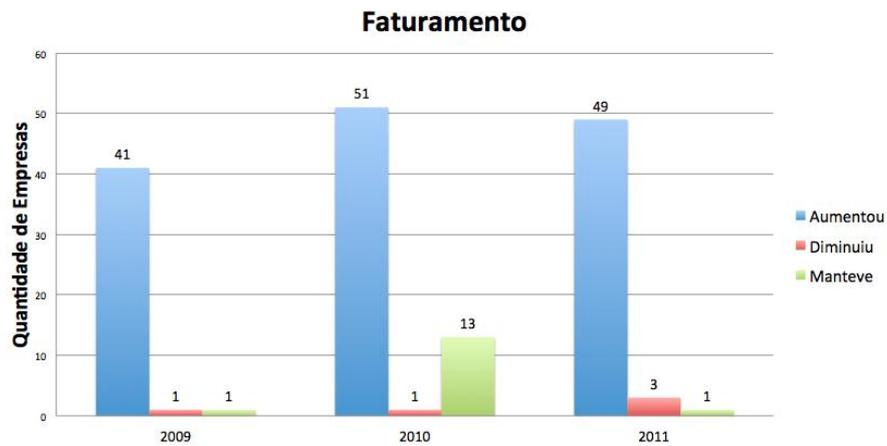


Figura 1 – Evolução do Faturamento das Empresas Certificadas MPS.BR

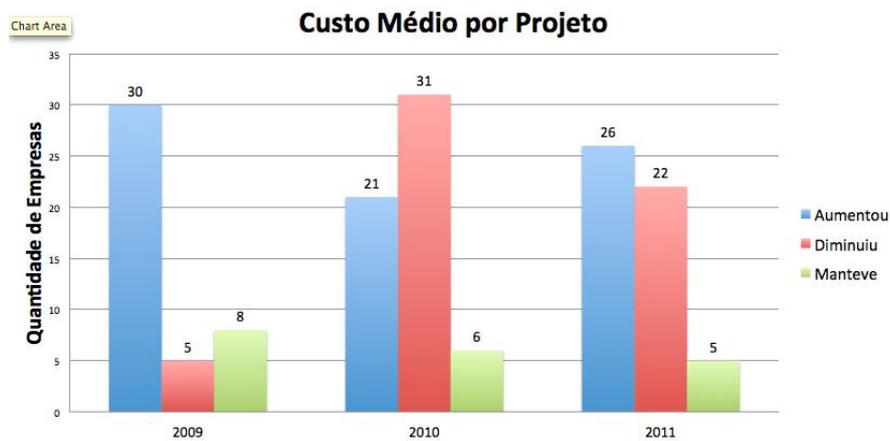


Figura 2 - Evolução do Custo dos Projetos das Empresas Certificadas MPS.BR

### Prazo Médio por Projeto

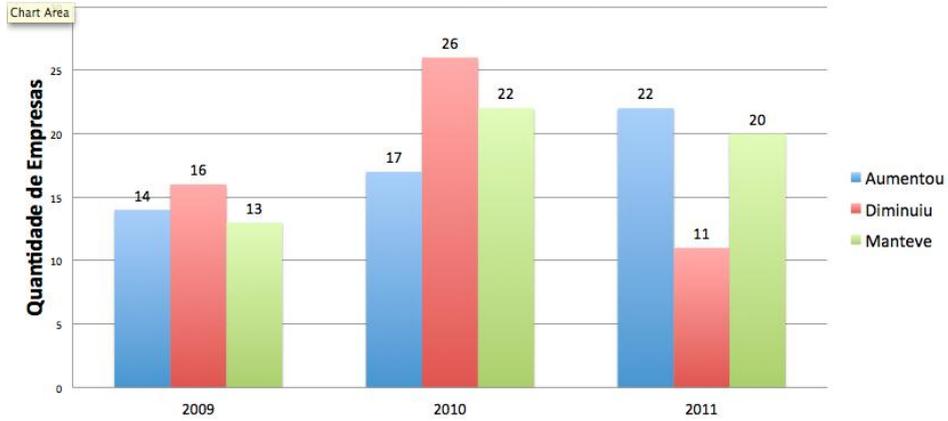


Figura 3 - Evolução do Prazo Médio dos Projetos das Empresas Certificadas MPS.BR

### Produtividade

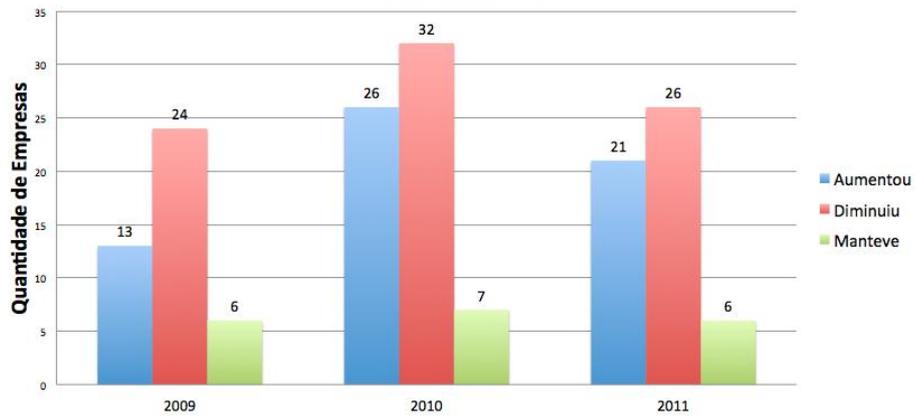


Figura 4 - Evolução da Produtividade das Empresas Certificadas MPS.BR

### Qualidade

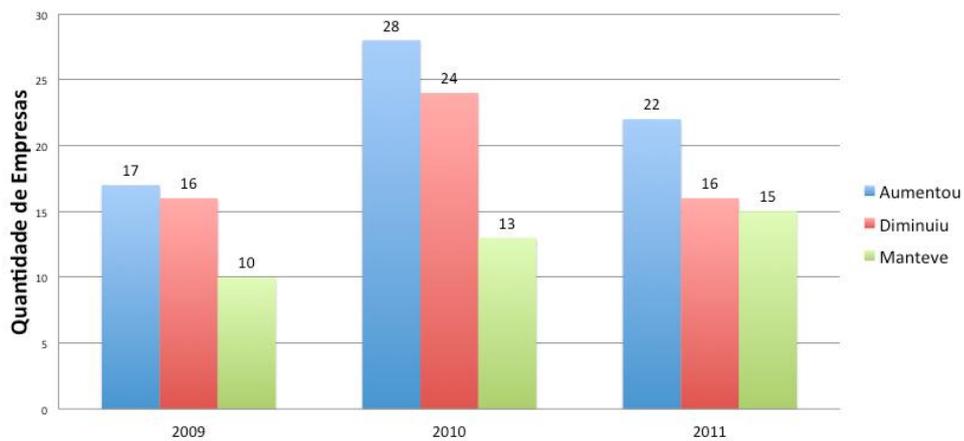


Figura 5 - Evolução da Qualidade dos Produtos das Empresas Certificadas MPS.BR

Outro dado coletado apenas nos iMPS 2010 e 2011 é o retorno do investimento (ROI) do programa MPS.BR dentro das organizações. A evolução deste indicador está apresentado na Figura 6 a seguir.

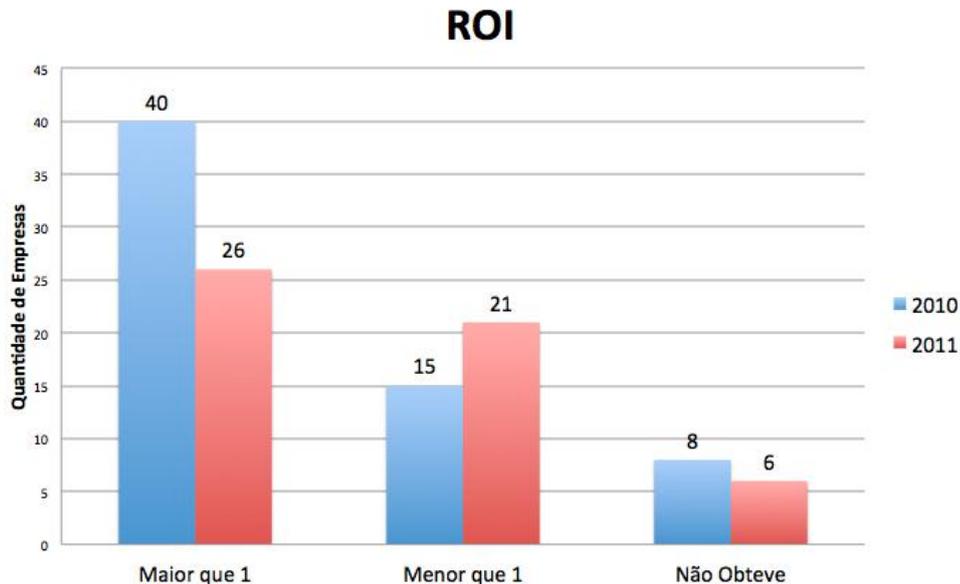


Figura 6 - Evolução da ROI das Empresas Certificadas MPS.BR

A partir destes indicadores podemos observar que em todos os anos, na maioria das empresas, o faturamento aumenta enquanto a produtividade diminui. Isso aponta para um cenário de que as empresas estão recebendo um volume maior de recursos enquanto entregam menos aos clientes. Isto pode indicar um cenário onde estas empresas parecem estar mais caras.

Se considerarmos o custo e tempo médio gastos por projetos, os indicadores de 2011 apresentam uma piora quando comparados àqueles de anos anteriores. Embora não possamos considerar que a piora seja uma tendência, os resultados de 2011 apontam para um cenário de projetos mais caros e demorados. Caso esta tendência seja mantida podemos afirmar que estes indicadores trazem um resultado negativo às empresas certificadas MPS.BR.

A qualidade também mostra um cenário peculiar. Embora haja um maior número absoluto de empresas que apresentam um aumento na qualidade, elas não são maioria. Em suma, não chega a 50% o total de empresas que aumentam a qualidade final do seu produto quando se submetem a certificação MPS.BR.

Se considerarmos que o MPS.BR foi criado com o intuito de aumentar a competitividade das empresas, estes indicadores não apresentaram um cenário de empresas que nitidamente produziram mais e melhor. Ao contrário, estas empresas se mostraram em sua maioria menos produtivas, com um maior custo por projeto e demorando mais em cada projeto, para assim ter como resultado geral que poucas delas apresentaram um acréscimo na qualidade final do seu produto.

Não há como fazer afirmações sobre a competitividade das organizações. Mas, é sugerido um cenário em que a maioria das empresas se tornam mais caras, produzem menos e demorem mais. Algo que, se confirmado, mostra uma perda de competitividade. E também, é válido ressaltar que apenas 60% das empresas em 2010, e 50% das empresas em 2011, obtiveram um retorno de investimento positivo com a adoção do programa, mesmo essas aumentando o faturamento.

Vale destacar que os indicadores do iMPS 2010 são bastante diferentes daqueles apresentados no iMPS 2011. Isso não significa que um universo completamente diferente de empresas foram entrevistados em ambas as pesquisas. Mas, certamente significa que algumas empresas entrevistadas em 2010 passaram mais um ano utilizando do seu processo MPS.BR, e responderam o questionário em 2011 com o uso do MPS.BR mais maduro dentro da organização. Isso significa dizer que o processo utilizado pela empresa no momento da certificação pode não ser mantido após um certo período de tempo, e conseqüentemente os indicadores organizacionais não se mantêm ao longo do tempo. O processo pode evoluir seguindo o caminho esperado pela melhoria de processo, mas o inverso também pode ser observado e o processo pode se mostrar menos eficiente do que no momento de sua criação.

Esse fenômeno de piora nos processos não é tão incomum. Coleman & Connor (2008) afirmam que na prática é observado que uma parte considerável das empresas que conseguem algum nível de maturidade, acabam diminuindo seu investimento em programas de melhoria de processo de software ao longo do

tempo, levando a uma erosão do processo. A erosão ocorre quando o processo atual é ignorado ou não utilizado, e a equipe volta a utilizar as práticas do processo anterior [Humphrey, 1989]. Esta realidade no próprio MPS.BR, mesmo não podendo ser verificada com exatidão, é percebida pelo fato de que ao menos 40 empresas não se re-certificaram. Ou seja, perderam seu nível de maturidade quando se verificou o histórico das avaliações do MPS.BR<sup>3</sup>. Esse número é bastante alto se considerarmos que só é necessário re-certificar as empresas que foram avaliadas até 2008, uma vez que a certificação MPS.BR tem validade de 3 anos. Então de um total de 123 empresas avaliações nesse período, pelo menos 40 não se submeteram a re-certificação (32%).

Não foram levantado todos os motivos que levam empresas a decidirem pela não re-certificação, contudo, dois deles são conhecidos: (i) a obtenção da certificação CMMI que possui grande reconhecimento no território nacional, portanto, as empresas preferem manter apenas a certificação CMMI, e (ii) por que as empresas não conseguem manter o processo concebido no momento da certificação, e acabam abandonando-o e decidindo não retomá-lo no momento da re-certificação. Apesar deste fenômeno, o número de avaliações MPS.BR está estável desde de 2009.

Segundo Colla e Montagna (2008), para as organizações de desenvolvimento de software, especialmente as pequenas e médias, o dilema de gestão é: Como justificar os investimentos necessários para melhorar o processo em um contexto de negócios em que concorrentes maiores, de escala global, desenvolvem ações semelhantes, aproveitando a maior estrutura, que é capaz de absorver melhor os efeitos da melhoria de processo? Ao mesmo tempo, elas devem considerar os concorrentes do mesmo tamanho que não introduziram melhorias significativas em seus processos e desfrutam de uma vantagem competitiva em curto prazo, para isso a margem de erosão deve ser a menor possível.

É exatamente neste contexto que se encontram 96% das empresas

---

<sup>3</sup><http://www.softex.br/mpsbr/avaliacoes/default.asp>

brasileiras. Elas necessitam se tornar competitivas no mercado, e para isso decidiram iniciar o programa de melhoria de processo baseado em modelos de maturidade, seja o CMMI ou MPS.BR.

Contudo o que vimos até agora é que na média estas empresas se tornam mais caras e menos produtivas. Além disso, existe a tendência de que algumas delas abandonem o processo utilizado para a certificação, retornando aos procedimentos mais simples realizados anteriormente. Mostra-se relevante, então, prover soluções alternativas, que sejam mais adequadas às empresas brasileiras, com o intuito de evitar esse abandono do processo. Uma alternativa seria a adoção de métodos ágeis.

Turk et al. (2002) afirma que os processos ágeis apoiam o "*management in the small*", onde os mecanismos de coordenação, controle e comunicação utilizados são aplicáveis a equipes pequenas e médias, e apoiam as pequenas a alcançar os seus resultados [Risinge Janoff, 2000].

Os benefícios das metodologias ágeis para pequenas equipes de trabalho, cujos requisitos mudam rapidamente, são notáveis [Siakas et al., 2005]. Abrahamsson (2005) afirma que a base do desenvolvimento ágil está em pequenas equipes que trabalham em ambientes co-localizados de desenvolvimento de software não críticos. Este ambiente apresentado por Abrahamsson (2005) é o mesmo da maioria das empresas brasileiras.

Outra motivação para a utilização de métodos ágeis é o resultado apresentado pelo *Chaos Manifesto*, apontando que projetos ágeis apresentam uma taxa de sucesso três vezes maior do que aquelas dos projetos cascata. Então, utilizar uma abordagem mais adequada a pequenas organizações e com um desempenho aparentemente melhor se mostra atrativo para as pequenas empresas da indústria nacional.

Considerando este cenário, foram propostas questões de pesquisa para avaliar a adoção desta solução alternativa para a melhoria de processo em pequenas organizações. As três questões de pesquisas propostas foram:

**QPI 1** *É possível executar programas de melhoria de processo de software baseados em métodos ágeis com o objetivo de certificação CMMI nível*

*dois e MPS.BR nível F?*

Esta questão de pesquisa é levantada pela necessidade de avaliar os programas de melhoria de processo que foram baseados em métodos ágeis, e que foram adotados por essas empresas. Avaliar se os mesmos são menos dispendiosos e mais adequados às necessidades de empresas pequenas e médias, que não conseguem manter o processo elaborado para atender os modelos de qualidade tais como o CMMI e MPS.BR. Usar métodos ágeis parece adequado para estas pequenas e médias empresas devido à natureza das abordagens ágeis. Mas, para esta pesquisa será considerado que a iniciativa de Melhoria de Processo baseada em métodos ágeis, identificada daqui para frente como MPS ágil, deve dirigir o programa desde o seu início. Ressaltamos que só foram investigadas empresas CMMI nível 2 e MPS.BR nível F, porque não foram encontradas empresas com níveis de maturidade mais altos que se enquadrassem nos critérios definidos por esta pesquisa, e que serão apresentados mais a frente.

**QP2** *ó Quais os benefícios observados por estas organizações ao se adotar a melhoria de processo baseada em métodos ágeis? E em longo prazo??*

Essa nova abordagem para melhoria de processo é mais rápida? Mais barata? Mais eficiente? Evita a erosão do processo? São as perguntas que esta RQ2 deve responder. Espera-se atingir resultados diferentes daqueles visualizados no iMPS.

**QP3** *ó Como a melhoria de processo ágil está relacionada à melhoria de processo prescritiva adotada pelo CMMI e MPS.BR.?*

Qual é o papel da melhoria de processo ágil dentro da organização depois que o CMMI ou o MPS.BR está instalado. O processo tende a seguir a rigidez do modelo de maturidade, ou continua apto a mudanças?

### 1.3. OBJETIVOS

O objetivo geral deste trabalho é avaliar os resultados da utilização de uma abordagem de MPS, com base em métodos ágeis, em empresas que desejam se certificar em modelos de qualidade. Esta avaliação deve identificar os benefícios

e as dificuldades de se adotar uma mentalidade ágil para a realização do MPS, e na medida do possível, comparar com os resultados obtidos pelas empresas brasileiras no relatório iMPS.

Para atingir estes resultados os seguintes objetivos específicos foram propostos:

- Explanar as diferenças entre programas MPS baseados nas mentalidades ágeis e nas tradicionais.
- Conduzir um estudo de caso para demonstrar a possibilidade de se obter uma certificação CMMI nível 2 e MPS.BR nível F utilizando esta mesma abordagem de melhoria de processo de software.
- Avaliar indicadores básicos das empresas, tais como qualidade, produtividade e custo, e compará-los com os resultados do iMPS.
- Identificar a longo prazo a erosão da parte ágil do processo, e a erosão do processo em relação ao CMMI/MPS.BR nas organizações.
- Identificar, através de entrevistas, que mudanças são percebidas nesta forma de MPS (ágil) em relação ao MPS baseado em modelos de maturidade promovido pelas abordagens tradicionais.

#### 1.4. CONTEXTO DE PESQUISA

Esta tese abrange quatro anos de estudos e 8 empresas envolvidas em nove avaliações das quais seis são CMMI nível 2/MPS.BR nível F e duas que abandonaram o processo para a obtenção do CMMI nível 2/MPS.Br nível F. Os critérios de inclusão das empresas na pesquisa foram (i) serem empresas de software de pequeno e médio porte, (ii) que estivessem em processo de implementação do programa de MPS, (iii) que fossem empresas que iniciaram estes programas usando métodos ágeis e (iv) que tivessem respondido os questionários.

O estudo de caso foi escolhido para esta pesquisa porque esta

metodologia e recomendada quando indivíduos, grupos, organizações e fenômenos sociais são investigados [Yin, 2002]. Além disso, estudos de caso são úteis ao avaliar os benefícios de métodos e ferramentas em empresas de software. [Kitchenham, Pickard e Pfleeger, 1995]. Estes tornam possível estudar um fenômeno contemporâneo dentro de seu contexto da vida real, especialmente quando os limites entre fenômeno e contexto não são claramente visíveis (Yin 2002). O que é verdadeiro para os estudos apresentados nesta tese.

É importante ressaltar que não é o objetivo deste trabalho fornecer os detalhes da implementação desta abordagem de MPS baseada em métodos ágeis, e que as empresas ficaram livres para adotar quaisquer estratégias. Por razões de confidencialidade os nomes das empresas não serão revelados. É importante enfatizar que estas empresas estão em diferentes regiões brasileiras. As 11 empresas identificadas são apresentadas a seguir:

**Empresa :** A empresa trabalha na área de automação comercial, tem 20 anos de mercado e cerca de 20 funcionários trabalhando na área de desenvolvimento de software. O produto desenvolvido por esta empresa está presente em 55% do território brasileiro e esta empresa tem o desenvolvimento de software como área fim, prestando serviços para empresas onde software é área meio. A empresa foi avaliada CMMI nível dois considerando o escopo de quatro projetos. Esta empresa também é certificada no nível F do modelo MPS.BR.

**Empresa :** Esta fábrica de software possui 11 anos de experiência e cerca de 40 funcionários que trabalham na área de desenvolvimento de software. Esta empresa atua no desenvolvimento de produtos de software em geral. Esta empresa tem o desenvolvimento de software como área fim e tem como clientes empresas públicas e privadas. Foi avaliada CMMI nível dois, considerando o escopo de cinco projetos. Esta empresa também obteve o nível F do MPS.BR.

**Empresa :** É uma fábrica de software com 16 anos de experiência e emprega cerca de 30 pessoas que trabalham na área de desenvolvimento de software. Esta empresa está voltada para o desenvolvimento de produtos de software em geral e tem o desenvolvimento de software como área fim. Esta empresa foi avaliada CMMI nível dois, considerando o escopo de cinco projetos. Esta empresa é

também certificada no MPS.BR nível F.

**Empresa** : É uma fábrica de software com 14 anos de experiência e cerca de 40 pessoas que trabalham na área de desenvolvimento de software. Esta empresa trabalha com transações financeiras eletrônicas. A empresa tem o desenvolvimento de software como área meio, em que o software é a ferramenta utilizada para a realização dos serviços. Esta empresa foi avaliada CMMI nível dois, considerando o escopo de cinco projetos. Também é certificada no MPS.BR nível F.

**Empresa** : É uma fábrica de software com 15 anos de experiência e cerca de 10 pessoas que trabalham na área de desenvolvimento de software na unidade selecionada para a avaliação. Esta empresa trabalha desenvolvendo produtos de software em geral sendo esta sua área fim. A empresa foi avaliada CMMI nível dois, considerando o âmbito de três projetos. É também certificada no MPS.BR nível F.

**Empresa** : É uma fábrica de software com 29 anos de experiência e cerca de 40 funcionários que trabalham na área de desenvolvimento de software em geral, onde este é área fim. Esta empresa está ligada ao ramo de saneamento, açúcar, saúde e gás. É certificada CMMI nível dois e também é certificada no MPS.BR nível F.

**Empresa** : É um órgão público com 20 anos de experiência e cerca de 20 pessoas que trabalham na área de desenvolvimento de software na unidade que será avaliada. Esta empresa não desenvolve software, que é a sua área fim, e trabalha apenas com terceirização. Por esta razão, a empresa iniciou o processo para ser avaliada no CMMI nível dois para a aquisição [Gallagher et. al., 2011] e o Modelo de Aquisição do MPS.BR (2011b). Esta foi uma das empresas que não prosseguiu para a avaliação.

**Empresa** : É uma empresa com 14 anos de experiência e cerca de 30 pessoas que trabalham em sua área de desenvolvimento de software na unidade que seria avaliada. Esta empresa trabalha com de digitalização de documentos e o desenvolvimento de software não é sua área fim. Esta empresa iniciou o projeto para a certificação, e após 12 meses o encerrou sem nenhuma certificação.

**Empresa :** Esta empresa possui cerca de 70 pessoas que trabalham nos projetos CMMI desta organização. Está alinhada para o desenvolvimento de software em geral e esta é sua área fim. Está certificada CMMI nível 2 e MPS.BR nível F, mas seus dados foram descartados por que a empresa não iniciou seu processo seguindo métodos ágeis, e assim não estava adequada aos critérios de inclusão propostos. O descarte de empresas nesta situação se deu para evitar o viés de considerar dados de organizações que não adotaram uma abordagem ágil de MPS e sim adotaram métodos ágeis como parte do processo de desenvolvimento. Isso é, utilizando métodos ágeis em abordagens de MPS tradicionais. Não é objetivo deste trabalho tratar de qualquer ambientes CMMI + Ágil, mas sim avaliar como empresas que se preocuparam em adotar uma abordagem ágil de MPS se comportaram em ambientes CMMI.

**Empresa :** Esta empresa possui cerca de 50 pessoas que trabalham nos projetos avaliados pelo CMMI desta organização. Não está orientada a nenhum segmento de negócios desenvolvendo software em geral. Está certificada no CMMI nível 3 e MPS.BR nível C, mas seus dados serão descartados por que a empresa não iniciou seu processo seguindo abordagens ágeis.

**Empresa :** Esta empresa possui cerca de 30 pessoas que trabalham nos projetos que foram avaliados pelo CMMI da organização. Está alinhada para o desenvolvimento de um produto interno e software é área fim. Está certificada no CMMI nível 5 e no MPS.BR nível A, mas seus dados serão descartados porque a empresa não iniciou seu processo seguindo métodos ágeis.

O Estudo de caso executado nestas empresas constitui a principal base empírica desta tese.

## 1.5. CONTRIBUIÇÃO E RESULTADOS ESPERADOS

**Contribuição da Tese:** O objetivo desta tese foi avaliar programas de MPS com fins de certificação CMMI/MPS.BR e que foram baseados em métodos ágeis. Esta avaliação observa se este tipo de MPS é apto a levar as empresas de fato a níveis de maturidade do CMMI ou MPS.BR. Este trabalho é o primeiro que se

propõe a avançar empresas que possuem estas características. Algumas contribuições desta pesquisa estão listadas a seguir.

- **Quantificação dos Resultados de Empresas que adotam Métodos Ágeis e CMMI:** Este é o primeiro relatório de pesquisa que apresenta dados quantitativos sobre o desempenho de empresas que utilizam métodos ágeis em programas de MPS para fins de certificação CMMI ou MPS.BR, e provê uma comparação do desempenho destas com as empresas brasileiras que se submetem a programas de MPS direcionados ao MPS.BR.
- **Avaliação do Processo de Longo Prazo:** Também não há estudos no Brasil que discutam o que acontece em longo prazo com os processos de empresas após a obtenção de qualquer tipo de certificação de qualidade de software. A erosão de processo em empresas de software brasileiras ainda não foi comentada em nenhuma pesquisa.
- **Identificação de que Elementos de Agilidade são encontrados nestes programas de MPS:** Embora haja diversos relatos sobre a utilização de métodos ágeis em ambientes CMMI e MPS.BR. Poucas delas são direcionadas a melhoria de processo. A maioria deles é relacionado ao nível de prática. Assim esse objetivo se resume a identificar quais aspectos da melhoria de processo ágil são encontradas neste meio prescritivo. Vale ressaltar que o conceito de agilidade adotado nesta tese é o mesmo sugerido por Highsmith e Cockburn (2001), em que a agilidade é definida como a capacidade de se adaptar as mudanças promovidas pelo ambiente.
- **Sugestão de um Perfil das Empresas Praticantes deste Modelo de MPS:** Esta pesquisa, a partir da realização de um *survey*, sugere um perfil de empresas que buscam o MPS ágil. Aqui temos só uma sugestão, ao invés de uma determinação, devido à pequena amostra de empresas envolvidas nesta pesquisa, e devido ao desconhecimento do tamanho da população de empresas que possuem certificações de qualidade e adotam métodos ágeis como abordagens de MPS,

- **Levantamento quantitativo da erosão do processo nestas empresas:**

Não foi encontrada nenhuma fonte de evidência quantitativa sobre a erosão de processo em nenhum tipo de MPS. Assim, esta pesquisa mesmo conseguindo dados de apenas três das onze empresas disponíveis, apresenta um levantamento de dados iniciais sobre como a erosão de processos ocorre dentro de um programa de melhoria de processo.

Além disso, este trabalho traz contribuições para as seguintes lacunas de pesquisa:

- **L2 (Moe, 2011)** ó Na lacuna de pesquisa L2 Moe (2011) alega que ainda é preciso verificar como ocorre o MPS em ambientes dinâmicos e turbulentos. Este trabalho apresenta dados sobre a identificação de características de MPS que ocorrem nestes ambientes orientados à mudança, bem como a observação desse comportamento a longo prazo;
- **L5 (Salo, 2007)** ó Nesta lacuna, Salo (2007) aponta que é necessário entender como se dá a integração do MPS ágil e MPS Tradicional. Este trabalho mostra como se dá a integração de MPS ágil tradicional no mesmo ambiente, entretanto, a abordagem de MPS deve iniciar a partir de métodos ágeis;
- **L6 (Salo, 2007)** ó Nesta lacuna, Salo (2007) apontou que era necessário investigar o comportamento do MPS em ambientes ágil + CMMI. Este trabalho apresenta uma junção do MPS ágil em uma abordagem CMMI e MPS.BR no mesmo ambiente para fins de certificação;
- **L9 (Pikkarainen, 2009)** ó Esta lacuna se refere a falta de pesquisas empíricas sobre métodos ágeis. Esta pesquisa apresenta a assimilação de práticas ágeis e de MPS ágil em organizações reais que possuem certificação CMMI e MPS.BR;
- **L10 (Santana, 2010)** ó Esta lacuna é referente aos elementos que envolvem toda a organização. Em se tratando do CMMI este elemento é as práticas genéricas. Este trabalho apresenta considerações sobre o relacionamento da adoção de programas de MPS ágil e as práticas genéricas do CMMI.

Até agora seis trabalhos relacionados a esta pesquisa foram publicados. A informação sobre os artigos é apresentadaa seguir nas Tabelas 3 e 4. Os artigos são apresentados no Apêndice E.

**Tabela 3–Relacionamento entre os trabalhos, principais resultados e as questões de pesquisa.**

Nº	QP	Resultados	Trabalhos
1	1	Métodos Ágeis apoiam CMMI	1, 2, 6
2	3	PráticasÁgeis percebidas em ambientes CMMI.	3, 4,
3	3	MPS Ágil em ambientes CMMI	2, 5, 6
4	2	Evolução do MPS Ágil em empresas CMMI.	4
5	2	CMMI não suporta a inovação dos processos ágeis	1

**Tabela 4- Relacionamento entre os trabalhos publicados e as empresas**

Nº	Trabalho	Empresa
P1	Agile Software Development and CMMI: What we do not know about Dancing with Elephants. [Santana et al., 2009a]	, e
P2	Software Process Improvement in Agile Software Development [Santana et al., 2010]	--
P3	Using Function Points in Agile Projects [Santana et al., 2011a]	, , e
P4	The Confidence of Agile Assessment Methods in the Context of Software Process Improvement [Santana et al., 2011b]	, , e
P5	Does Agile Development Fit in the Actual Context of Software Quality? [Santana et al., 2012]	--
P6	Utilização de Práticas Genéricas do CMMI para apoiar a utilização de Metodologias Ágeis. [Santana et al, 2009b].	

**P1:**Santana, C., Gusmão, C., Vasconcelos, A. & Pinheiro, C. (2009). Agile and CMMI what do we know about dancing with elephant. In international conference on agile process and extreme programming (XP 2009).

**Relevância para a Pesquisa:** O trabalho apresentou relatos de três empresas especificadas na Tabela 4. Neste trabalho foi possível ver o apoio que métodos ágeis trazem para o CMMI, mas a perda de agilidade das empresas estudadas foi percebida.

**P2:**Santana, C., Cervino, M., Caetano, D. & Gusmão, C. (2010). Software Process Improvement in Agile Software Development. In 1º Workshop Brasileiro de Metodologias Ágeis (WBMA-2010).

**Relevância para a Pesquisa:** O artigo apresenta um mapeamento doMPS ágil. A principal contribuição do estudo foi levantar novos aspectos que devem ser considerados além da reciclagem teórica sobre o Tema.

**P3:**Santana, C., Leoneo, F., Gusmão, C. & Vasconcelos, A. (2011). Using Function Points in Agile Projects. In international Conference on Extreme Programming and Agile Process in Software Engineering (XP 2011).

**Relevância para a Pesquisa:** O trabalho apresenta um estudo de caso, em que as empresas devem utilizar os pontos de história e pontos de função no mesmo ambiente. A principal contribuição do estudo foi identificar como evitar que uma determinada empresa (Empresa J)perdesse agilidade.Outra contribuição é que a mesma abordagem foi replicada para a empresa M, e em ambos os casos a medição e análise do CMMI foi mantida.

**P4:**Santana, C., Caetano, D., Gusmão, C. & Vasconcelos, A. (2011). The Confidence of Agile Assessment Methods in the Context of Software Process Improvement. In Agile 2011 Conference (Agile 2011).

**Relevância para a Pesquisa:** O trabalho apresenta um estudo de caso em que as empresas avaliam o seu próprio processo utilizando o Nokia Test.A evolução deste teste foi comparada com a evolução dos principais indicadores organizacionais. Em alguns casos, o resultado final do *Nokia Test* foi incorporado à base de métricas de organização como indicador de produtividade e qualidade.

**P5:**Santana, C., Filho, H., Gusmão, C.& Vasconcelos, A. (2012). Does Agile Development Fit in the Actual Context of Software Quality?In Agile India 2012 Conference.

**Relevância para a Pesquisa:** O artigo apresenta uma investigação teórica sobre as abordagens de qualidade em engenharia de software. Este trabalho auxilia na compreensão das abordagens de melhoria de processos de software tradicionais e ágeis quando estas abordagens são aplicadas no desenvolvimento de software.

**PO:** Santana, C., Gusmao, C., Vasconcelos, A. & Rouiller, A. (2009). Utilização de Práticas Genéricas do CMMI para apoiar a utilização de Metodologias Ágeis. In Conferência Íbero-Americana de Engenharia de Software (IDEAS 2009).

**Relevância para a Pesquisa:** Este trabalho apresenta como as práticas genéricas do CMMI podem auxiliar a implantação de uma cultura ágil em organizações que desejam obter a certificação CMMI.

## 1.6. ESTRUTURA DA TESE

Além deste capítulo introdutório, esta tese possui mais quatro capítulos conforme mostrado na Tabela 5.

**Tabela 5-Estrutura do Documento**

Capítulo	Conteúdo
2 ó Referencial Teórico	Este capítulo é composto por uma breve introdução para o fundo teórico da pesquisa apresentada nesta tese.
3 - Metodologia	Neste capítulo é mostrada a metodologia do trabalho que é baseada em estudo de caso.
4 - Análise dos Resultados	Neste capítulo é apresentada a análise dos resultados obtidos durante a pesquisa.
5 - Conclusão	Resumimos as contribuições da tese, discussão dos resultados, ameaças à validade, impactos na indústria e na academia e trabalhos futuros.
Elementos Pós-Textuais.	Anexos, Apêndices e Referências Bibliográficas virão após a conclusão

## 2. Referencial Teórico

Este capítulo apresenta o referencial teórico relacionado à melhoria de processo de software ágil em ambientes CMMI/MPS.BR, que estabelece a base para esta pesquisa.

### 2.1 MELHORIA DE PROCESSO DE SOFTWARE

Pesquisas sobre melhoria de processo de software são motivadas pela suposição comum de que a qualidade do processo está diretamente relacionada com a qualidade do produto de software desenvolvido. O objetivo da melhoria de processo de software é, portanto, aumentar a qualidade do produto e reduzir o tempo de mercado e custos de produção [Fuggetta, 2000].

Uma das características do MPS tradicional é sua ênfase na melhoria contínua dos processos de desenvolvimento organizacional de software, em termos de: desempenho, estabilidade, capacidade e confiança. Muitas vezes, as abordagens existentes para MPS parecem reforçar os objetivos de negócio e as necessidades de melhoria do processo de desenvolvimento de software da organização. As iniciativas tradicionais de MPS também são estritamente controladas e gerenciadas pelo nível organizacional [Boehm e Turner, 2003].

A estrutura da melhoria de processo de software é originada no ciclo de *Shewhart-Deming* que prega os seguintes passos: estabelecer um plano de melhoria; implementar o novo processo; medir os resultados das mudanças do processo; e analisar o efeito das mudanças implementadas. Este ciclo é também chamado de PDCA ou *Plan- Do - Check - Act* [Gorschek e Wohlin, 2004; Pettersson et al., 2008].

Uma característica comum das iniciativas de melhoria de processo de software tradicional é a sua abordagem para identificar os processos, *ōto-be-improved*. Esta abordagem prega que os processos reais devem ser comparados com um conjunto de boas práticas. Em caso de divergências, devem ser identificadas as oportunidades de melhoria, e eliminadas as diferenças entre o

processo real e o conjunto de boas práticas providas pelos modelos [Siakas et al., 2005]. Esta abordagem é comumente referida como melhoria *top-down* ou prescritiva [Pettersson et al., 2008]. Em oposição a este conceito existem abordagens de melhoria de processo definidas como *bottom-up* ou indutivas.

Em contraste com essas abordagens prescritivas, os métodos indutivos de MPS apresentam uma abordagem *bottom-up* baseado que deve ser realizado, em termos de melhorias organizacionais, face a compreensão aprofundada da situação atual do processo. A ideia é fundamentar a melhoria do processo mediante experiências na execução de projetos. Não existe uma avaliação inicial formal ou comparação com um conjunto pré-definido de práticas. Em vez disso, metas quantificáveis são definidas e, a partir delas, melhorias são escolhidas [Pettersson et al., 2008].

Salo (2008) mapeia seis elementos diferentes identificados no contexto de MPS tradicional:

- Os modelos organizacionais para melhoria contínua;
- Os processos padrão de avaliação;
- Adaptação de processos;
- Implantação de processos;
- Medição do processo;
- Conhecimento e aprendizagem organizacional.

**Discussão da Seção:** A Melhoria de Processo de Software vem sendo adotada por grande parte da indústria de software como conceito para a garantia da qualidade do produto final de software. Existem duas abordagens mais comuns para o MPS (i) prescritiva, em que um modelo de referência é utilizado como guia para a melhoria de um determinado processo, e (ii) indutiva, aquela que a organização identifica as oportunidades de melhoria e conduz a melhoria desejada baseada na própria experiência.

O modelo tradicional do desenvolvimento de software prefere abordagens do tipo prescritivas, sendo percebido pelo grande número de normas, padrões e modelos existentes no mercado. Deste modo percebemos que empresas que são

orientadas a planejamento (tradicionalis) tendem a seguir modelos na montagem de seus processos.

Por último são identificados por Salo (2008) temas comuns no MPS que são problemas recorrentes em muitas dessas empresas, que são: (i) modelos em si, e em que contexto eles devem ser utilizados; (ii) como o processo será avaliado diante do modelo; (iii) como o modelo será adaptado ao processo da empresa; (iv) como ocorrerá a implantação desse processo na empresa; (v) como esse processo será medido; e por último, (vi) como a experiência obtida a partir da utilização do processo será absorvida pela empresa.

## 2.2 MELHORIA DE PROCESSO DE SOFTWARE BASEADA EM MODELOS DE MATURIDADE

Uma das definições do termo modelo de maturidade foi dada por John Schlichter, criador do *Organizational Project Management Maturity Model* ou OPM3 [PMI, 2003]. Ele afirma que um modelo de maturidade é um conjunto de boas práticas que são potencializadas dentro de uma organização na medida em que a mesma o utiliza. A maturidade em si representa o aprimoramento da utilização destas boas práticas, alinhadas aos negócios da organização.

O CMMI é uma abordagem consolidada, influente e frequentemente estudada para MPS. O modelo descreve práticas e metas para as áreas de processo de desenvolvimento de software, e tem estruturas para medir a conformidade das organizações com os seus objetivos e práticas [Staples e Niazi, 2007; Tarawneh et al., 2011.].

De fato, *Capability Maturity Model Integration* (CMMI) e o modelo de Melhoria de Processo do Software Brasileiro (MPS.BR) são modelos de maturidade de melhoria de processo software [Chrissis et al., 2011; MPS.BR, 2011a].

Em ambos os modelos são sugeridos níveis de maturidade para as organizações, e cada nível de maturidade indica o estágio de organização de uma empresa. O CMMI sugeriu em seu modelo cinco níveis de maturidade se

baseando no modelo de quantidade de Crosby (1979), sendo eles: (1) **Executado** que indica que a organização executa o processo, (2) **Gerenciado** que indica que a organização gerencia os seus processos, (3) **Definido** que indica que a organização já tem um processo padrão e uma forma de adaptar esse processo para cada situação, (4) **Gerenciado Quantitativamente** que indica que o processo da organização é avaliado em algumas áreas de maneira quantitativa, e por último (5) **Em otimização** que indica que as melhorias do processo são realizados com base em dados quantitativos [Chrissis, et al., 2011].

O MPS.BR possui um conceito parecido e mais gradual do que o CMMI. São sete níveis de maturidade onde os níveis G+F do MPS.BR equivalem ao nível 2 CMMI; os níveis E+D+C equivalem ao nível 3 do CMMI, o nível B equivale ao nível 4 do CMMI, enquanto o A equivale ao nível 5 do CMMI [MPS.BR, 2011a].

As boas práticas dos modelos estão divididas por áreas de conhecimento chamadas de áreas de processo (CMMI) ou processos (MPS.BR). Estas correspondem às grandes áreas da engenharia de software. Os dois modelos têm como referência as áreas de processo da ISO 12207 [ISO, 1995]. O CMMI atualmente aborda 22 áreas de processo, enquanto o MPS.BR aborda 25 processos [Chrissis, et al., 2011; MPS.BR, 2011a].

Estes modelos também estão divididos em elementos que atingem somente uma única área ou atingem todas as áreas. Por exemplo, criar um mecanismo de rastreabilidade entre requisitos é um elemento que atinge somente o processo de gerência destes, enquanto que o planejamento de áreas irá atingir todas as áreas, uma vez que haverá um planejamento sobre como gerenciar os requisitos; sobre como gerenciar os projetos, como gerenciar a configuração, entre outras áreas.

No CMMI estes elementos que atingem apenas uma área de processo em particular são chamados de práticas específicas, pois, aquela boa prática é específica daquela área. As práticas que atingem todas as áreas de processo são chamadas de *práticas genéricas*, que servem para todas as áreas de processo [Chrissis et al., 2011].

NO MPS.BR existe o mesmo conceito, só que, chamados de Resultados Esperados para aquelas práticas que só atingem um processo, e Atributos de Processo para aquelas que atingem todas as áreas [MPS.BR, 2011a].

Em ambos os modelos os níveis de maturidade do processo são atingidos de acordo com a utilização das práticas que são comuns a todas as áreas. No CMMI isto é chamado de nível de capacidade, que representa o nível de utilização das práticas genéricas por uma organização. Para o MPS.BR, embora não haja um nível de capacidade definido, existem sete níveis de Atributos de Processo um para cada nível de maturidade neste modelo. Neste caso o modelo de melhoria de processo adotado é prescritivo, uma vez que a organização tenta melhorar os seus processos seguindo as recomendações destes modelos, e se assume que quanto maior a maturidade, melhores são os resultados da organização [Rainer e Hall, 2002].

Staples e Niazi (2007) conduziram uma revisão sistemática e com base em mais de 80 trabalhos, e a partir dele levantaram as principais razões para a adoção da melhoria de processo de software baseada em modelos. Os motivos estão mostrados na Tabela 6.

**Tabela 6. Motivos para a adoção de programas de MPS baseados em modelos de Maturidade [Staples e Niazi, 2007].**

Motivo	Percentual
1 - Qualidade de Software	65,7%
2 ó Tempo de Desenvolvimento	54,5%
3 ó Custo do Desenvolvimento	54,5%
4 - Produtividade	41,3%
5 ó Viabilidade do Processo	30,1%
6 ó Demanda do Cliente	21,0%
7 ó Vantagem Competitiva	20,1%
8 ó Melhoria do Processo	17,5%

Em outra revisão sistemática, Staple et al. (2007) identificaram também os motivos para **não** se adotar este tipo de abordagem de melhoria de processo. Os resultados estão mostrados na Tabela 7.

**Tabela 7. Motivos para a não adoção de programas de MPS baseados em modelos de Maturidade [Staples et al., 2007].**

Motivo	Percentual
1 ó Organização muito pequena	43,0%
2 ó Alto Custo	35,0%
3 ó Falta de Tempo	25,0%
4 ó Utilizando outras abordagens para MPS	20,0%
5 ó Falta de Benefícios Claros	10,0%
6 ó Benefícios prometidos não são prioritários	8,0%
7 ó O Cliente não demanda	5,0%
8 ó Não é aplicável aos projetos	5,0%

No Brasil existem poucos estudos relatando como o CMMI é utilizado. O estudo mais antigo sobre a adoção de modelos de qualidade no Brasil foi realizado por Weber e Pinheiro (1995), apontando que 9 empresas brasileiras eram certificadas no modelo SW-CMM. Kival e Amaral (1999) e Kival et al. (2000) mostraram o crescimento gradual do uso deste modelo até chegar a 44 empresas certificadas no ano de 1999.

Veloso et al. (2003) aponta que mesmo com esse crescimento, o Brasil tem uma indústria de software menos orientada ao CMM do que a Índia, e que até aquele momento ainda não existia nenhuma empresa certificada CMM/CMMI nível 5 no Brasil.

Barbará et al. (2003) relata que no Brasil a utilização de tal modelo não se transformou em resultados positivos perceptíveis para as organizações que o adotaram. Os autores acreditam que apenas as empresas com nível de maturidade cinco seriam mais beneficiadas com a utilização do modelo, o que não era o caso da indústria brasileira.

Em 2005 surge a iniciativa do MPS.BR, e ainda neste ano Rocha et al. (2005) apresentam um conjunto de fatores de sucesso e insucesso em implementações de programas de MPS baseados no CMMI e MPS.BR nas empresas brasileiras. Os resultados estão na Tabela 8 e 9, respectivamente.

É importante ressaltar que nestes questionários as empresas poderiam votar em mais de uma opção.

**Tabela 8. Fatores de Sucesso na Implementação do MPS.BR e CMMI em empresas Brasileiras [Rocha et al., 2005].**

Fatores de Sucesso	Votos (Empresas)
1 ó Comprometimento da empresa	29
2 ó Apoio ferramental	19
3 ó Motivação	17
4 ó Estratégia de implementação	16
5 ó Treinamento	16
6 ó Acompanhamento dos processos	12
7 ó Recursos disponíveis	11
8 ó Competência dos implementadores	7

**Tabela 9. Fatores de Insucesso na Implementação do MPS.BR e CMMI em empresas Brasileiras [Rocha et al., 2005].**

Fatores de Insucesso	Votos (Empresas)
1 ó Competência da equipe da empresa	23
2 ó Cultura organizacional	21
3 ó Estratégia de implementação	18
4 ó Comprometimento da empresa	15
5 ó Acompanhamento dos processos	11
6 ó Estrutura da empresa	12
7 ó Recursos disponíveis	11
8 ó Recursos financeiros	10

Devemos ressaltar que o crescimento de avaliações MPS.BR é notório. Enquanto em 2005 foram apenas 5 avaliações; em 2011 foram 71 avaliações.

**Discussão da Seção:** A Melhoria de Processo de Software Baseada nos Modelos de Maturidade é uma estratégia prescritiva de MPS, em que as empresas irão se adequar ao nível de maturidade desejado nestes modelos. Podemos perceber inicialmente que estes modelos afetam toda a estrutura de desenvolvimento da

organização, uma vez que englobam boa parte das áreas envolvidas no desenvolvimento de software.

A maturidade tão ressaltada nesses modelos não se trata, conceitualmente falando, do nível de qualidade do produto final, nem do processo adotado, mas sim da melhoria do processo de software empregada pela organização. Em outras palavras, significa que a organização estará experiente em executar e alterar (realizar a melhoria) do seu processo definido, bem como avaliar se a melhoria proposta surtiu efeito.

Este conceito de MPS é baseado no conceito geral de garantia da qualidade, que tem como lema *“Fazer certo na primeira vez”*. Isto significa que se a organização descobriu uma forma de realizar o seu trabalho utilizando o CMMI uma vez (primeira vez), a organização irá repeti-la e atualizá-la (melhorá-la) com o passar do tempo.

Novamente, é fácil perceber o porquê deste modelo até 2003 não ter trazido benefícios perceptíveis para as empresas brasileiras. Se o processo proposto for pesado e falho, a maturidade só indicaria que a organização está experiente em rodar esse processo. Se, ao invés de melhorias, forem propostas mudanças que tragam pioras, o processo continuará sendo um problema para a organização que o adotou.

Outro problema grave que talvez tenha sido percebido pelas empresas brasileiras, é que as mudanças propostas em seus processos dificilmente terão como alvo seus objetivos de negócio, e sim a aderência a modelos de maturidade.

A realidade atual não é tão diferente daquela apresentada por Barbará et al em 2003, pois, como visto na Seção 1.2, as empresas brasileiras não apresentaram melhorias significativas com a adoção do MPS.BR..

### 2.3 MENTALIDADE ÁGIL

Os métodos ágeis foram formalizados em 17 de Fevereiro de 2001, por 17 líderes de desenvolvimento que propuseram métodos até então conhecidos como

oieveso. Esse encontro verificou pontos comuns dessas abordagens, e resultou em um acordo que possuía quatro níveis [Koch, 2005], daqui em diante este acordo será tratado nesta tese como acordo ágil.

O *primeiro nível* considera a necessidade da existência de métodos construídos para responder a mudanças durante o projeto de software. O termo *Ágil* foi adotado para esses métodos, pois, o termo *leve* não era apropriado, principalmente para certos projetos que teriam restrições em empregar metodologias leves, mas, que ainda assim, poderiam precisar de agilidade.

O *segundo nível* do acordo foi a publicação do Manifesto Ágil<sup>4</sup>, manifesto esse composto por quatro valores centrais que são aplicados por todas as metodologias que assim se denominam:

- Indivíduos e iterações mais do que processos e ferramentas;
- Software funcionando mais do que documentação extensa;
- Colaboração do cliente mais do que negociação de contratos;
- Responder a mudanças mais do que seguir um plano.

Ainda que haja valor nos termos à direita, são mais valorizados pelos métodos ágeis os termos à esquerda.

O *terceiro nível* é composto por um conjunto de doze princípios. Neste nível os valores são fornecidos com detalhes mais concretos do que os quatro valores do nível anterior. Estes princípios são:

- Nossa maior prioridade é prover a segurança do cliente através de entregas antecipadas, e de forma contínua, de software de valor ao cliente;
- Mudanças de requisitos são bem-vindas, mesmo em fases tardias do desenvolvimento. Processos ágeis abraçam mudanças para promover a vantagem competitiva do cliente;
- Entregar software funcionando frequentemente, entre poucas semanas a poucos meses, dando preferência à escala de tempo mais curta;
- Pessoas de software e de negócios devem trabalhar juntas diariamente

---

<sup>4</sup> [www.agilemanifesto.org](http://www.agilemanifesto.org)

atraves do projeto;

- Construa projetos em torno de indivíduos motivados, dê-lhes o ambiente e apoio que necessitem, e confie neles para que o trabalho seja feito;
- A forma mais eficiente e efetiva de trazer informação para o time, e dentro do mesmo, é a comunicação face a face;
- Software funcionando é a primeira métrica de progresso;
- Processos Ágeis promovem um ritmo sustentável. Patrocinadores, desenvolvedores e usuários devem estar aptos a manter o ritmo indefinidamente;
- Atenção contínua à excelência técnica e bons projetos ajudam a agilidade;
- Simplicidade: a arte de maximizar a quantidade de trabalho que não deve ser realizado é essencial;
- As melhores arquiteturas, requisitos e projetos surgem de times auto-organizados;
- Em intervalos regulares, o time deve refletir sobre como se tornar mais efetivo, e então ajustar seu comportamento de acordo com a reflexão.

O quarto e *último nível* deste acordo seria aquele no qual cada abordagem ágil deveria definir a si mesma. Alguns exemplos de abordagens ágeis são a *extreme programming* (XP) [Beck, 2004] e Scrum [Schwaber e Beedle, 2001].

A mentalidade ágil começou ser discutida por Ambler (2009), como uma resposta àqueles que estavam solicitando provas e evidências de que os métodos ágeis realmente funcionavam.

Até aquele momento, a comunidade ágil não tinha se preocupado em definir o que é ser ágil ou não, e muito menos com algum modelo de avaliação sobre ser ágil.

Com a rápida popularização de XP e Scrum, foi percebido que muitos dos problemas enfrentados por times ágeis decorriam da falta da cultura de mudança proposta pelos métodos ágeis. Ambler (2009) aponta que muitas

empresas começaram a adotar XP e Scrum indiscriminadamente, sem alinhá-los aos objetivos de negócio da organização. Como resposta, Ambler apresentou uma imagem (Figura 7) para iniciar sua definição do que é ser ágil.

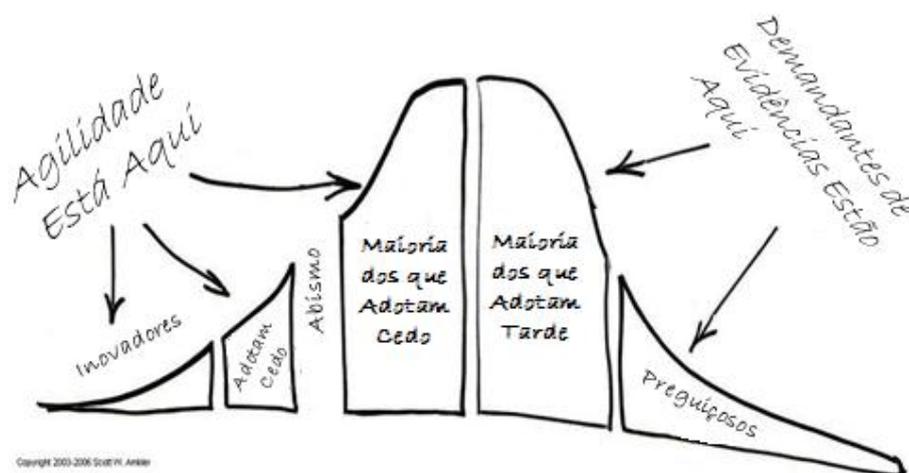


Figura 7 – Quando organizações podem se definir ágeis [Ambler, 2009]

Ambler (2009) afirma que empresas ágeis são aquelas que utilizam um conceito novo assim que ele é lançado; se ele funciona é continuado, senão é abandonado pelas organizações. As empresas que são mais tradicionais só costumam adotar tecnologias que estão consolidadas no mercado e ainda assim tentam adaptá-las o máximo possível à sua cultura organizacional, mesmo que os princípios envolvidos sejam diferentes.

Highsmith (2011) apresentou um seminário na rede sobre o parâmetro de fazer ágil e começar a ser ágil. Neste seminário foi apresentada uma crítica a empresas que estavam adotando métodos ágeis como ferramenta, mas, se esqueciam do princípio ágil de abraçar mudanças. Ou seja, as empresas estavam indo direto ao 4º nível do acordo ágil sem planejar dentro de sua cultura organizacional como os outros três níveis seriam implementados.

O mesmo tema também foi citado por Kruchten (2011) em um artigo especial sobre os 10 anos das metodologias ágeis. Neste artigo, Kruchten afirma que um dos grandes problemas mal resolvidos no uso de métodos ágeis é a definição do contexto de ser ágil.

Mais recentemente, Williams (2012) mostrou que 68% dos times ágeis acham importante seguir os níveis mais altos deste acordo, pois, se a empresa

realmente deseja ser ágil, esses são os níveis em que elas devem se basear antes de adotar qualquer tipo de prática. Entretanto, o que é percebido na prática é justamente o oposto, as empresas adotam os métodos e técnicas (4º nível do acordo ágil) sem observar os níveis acima.

Highsmith (1999) promove uma comparação intrigante entre a cultura orientada a mudanças e a cultura tradicional. Ele afirma que a indústria tradicional tem uma característica otimizadora. Que é quando uma empresa define sua forma de trabalhar, repete essa fórmula de trabalho, e otimizações são sugeridas para tornar o produto final mais barato, com maior qualidade ou mais mesmo rápido de ser produzido.

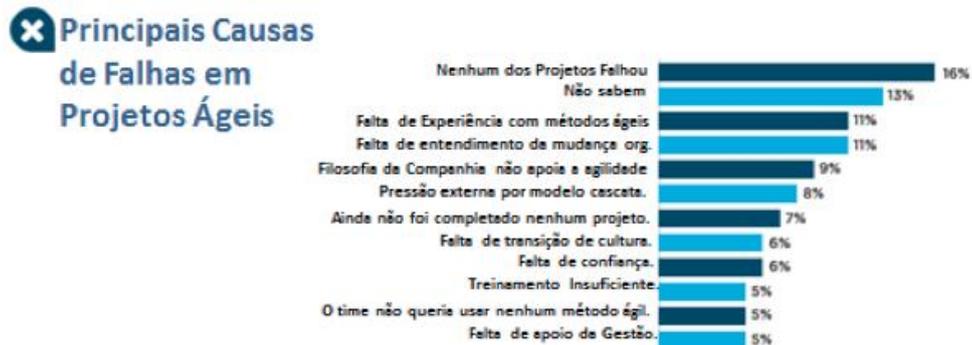
Por outro lado, a mudança da otimização para a adaptação em uma organização ou projeto exige uma profunda mudança cultural. Primeiro, devemos abandonar o nosso senso de arrumação e organização. Claro que existem lugares onde é necessário ordem, mas em geral o ritmo das mudanças do mundo dos negócios se opõe a este desejo.

**Discussão da Seção:** A mentalidade ágil definida a partir do acordo ágil traz à tona elementos completamente distintos daqueles vistos pelas abordagens tradicionais. A definição de agilidade utilizada em toda esta pesquisa respeita o primeiro nível do acordo ágil pode ser assim expressa: **Agilidade é a capacidade de adaptação das organizações face as mudanças ocorridas no ambiente de negócios.**

Este conceito é refletido na cultura, estrutura, processos, valores, práticas e técnicas de uma organização e interfere diretamente na forma de trabalho das pessoas envolvidas nos projetos. A grande questão é a respeito da cultura da otimização vista nas abordagens tradicionais *versus* a cultura da adaptação. Claramente se pode verificar que a melhoria de processo de software baseada em modelos de maturidade remete à cultura de otimização onde é definido um processo, e que este processo é executado repetidas vezes e melhorado para que a empresa tenha um melhor desempenho. Sendo que esta visão destoa o que a cultura de adaptação (ágil) prega.

Esta diferença de pensamento já foi identificada nas discussões realizadas nas grandes conferências e fóruns que tratam de métodos ágeis. Muitas organizações começam a utilizar Scrum e XP e rotulam a si próprias de ágeis, contudo elas esquecem os níveis mais altos do acordo ágil e passam a ter problemas com o choque de cultura. Em sua cultura otimizada, esta organização tenta realizar a melhoria para resolver este problema e propõe uma solução que não está de acordo com a mentalidade ágil, e a disponibiliza como solução híbrida.

Esta parcela de empresas híbridas já alcançou um número considerável quando consideramos o universo de empresas que se intitulam ágeis. O questionário sobre a situação do desenvolvimento ágil trazido pela Versionone (2011) mostra que 40% das empresas que falharam em projetos ágeis colocam a responsabilidade da falha em motivos culturais e de mentalidade. Pode-se visualizar o resultado original da Versionone (2011) na Figura 8 a seguir.



**Figura 8– Por que projetos ágeis falham [Versionone, 2011]**

O questionário da Versionone mostra que as falhas decorridas em projetos ágeis ocorrem em parte pela falta de alinhamento estratégico aos negócios da organização. Isso é percebido ao observarmos cinco dos motivos: (i) falta de entendimento da mudança organizacional, (ii) filosofia da companhia não apoia agilidade, (iii) pressão externa pelo modelo cascata, (iv) falta de transição de cultura, e (v) falta de apoio da gestão.

## 2.4 MELHORIA DE PROCESSO DE SOFTWARE ÁGIL.

Segundo Moe (2011) existem diferenças fundamentais entre o MPS no desenvolvimento de software tradicional e ágil. Em primeiro lugar, enquanto o MPS nas abordagens tradicionais é prescritivo, colocando normas para a forma como os indivíduos, equipes, organização devem operar; no desenvolvimento ágil é preferida uma abordagem indutiva, dirigindo-se à melhoria do comportamento e gestão de práticas de desenvolvimento de software dentro de times [Lycett et al., 2003].

No desenvolvimento ágil, processos não são produtos, mas sim práticas que evoluem de forma dinâmica dentro da equipe, e como esta se adapta às circunstâncias particulares [Aaen, 2008]. São incentivados o autogerenciamento e a comunicação face a face, para encontrar a melhor forma de desenvolver software.

Salo (2007) e Moe (2011) afirmam que existem três dos doze princípios do Manifesto Ágil que dão suporte a MPS ágil:

- O método mais eficiente e eficaz de transmitir informação dentro de uma equipe de desenvolvimento é a conversa face a face;
- Em intervalos regulares a equipe deve refletir sobre como se tornar mais efetiva, em seguida, ajustar seu comportamento apropriadamente;
- As melhores arquiteturas, requisitos e projetos emergem de equipes auto-organizadas.

Outra diferença entre as abordagens é que a MPS tradicional, muitas vezes, enfatiza a melhoria contínua dos processos organizacionais para projetos futuros, enquanto os princípios do desenvolvimento de software ágil estão na iteratividade. É que a melhoria dos projetos é realizada quando estes estão em andamento. Ciclos curtos de desenvolvimento fornecem iterações contínuas e rápidas de aprendizagem, interativas tanto para melhorar os processos quanto para testar as melhorias [Salo, 2007; Moe 2011].

Adotar uma abordagem ágil implica em mudar muitos aspectos da organização, incluindo sua estrutura, cultura e gestão [Nerur et al, 2005.]. Neste caso, são exigidas significativas mudanças organizacionais que levam muito

tempo [Pyzdek 1992; Vinekar et al, 2006. ].

Salo (2007) argumenta que devido às diferenças fundamentais entre o desenvolvimento de software tradicional e ágil, há uma necessidade de definir novos mecanismos de MPS para o desenvolvimento de software ágil. Salo sugere um processo iterativo de melhoria para a realização de MPS dentro dos times, pois a principal crítica contra o MPS tradicional é que nele existe um excesso de burocracia que restringe a inovação [Boehm & Turner, 2003]. O desenvolvimento de software tradicional enfatiza as negociações iniciais do contrato, em que os requisitos relacionados a custo e tempo são fixos eo produto final será entregue no final do ciclo de vida do projeto. Nesta forma de desenvolvimento de software tradicional existe uma tendência ao uso de documentação extensa e monitoramento quantitativo do processo, o que não é apoiado pelos métodos ágeis [Salo 2007].

Hoje em dia a personalização e flexibilidade nos métodos de trabalho, o envolvimento do cliente e participação do usuário são essenciais [Siakas et al., 2005]. Métodos Ágeis abordam estas questões de forma que os próprios times encontrem seus processos para a utilização de seus próprios conhecimentos e necessidades de documentação [Siakas et al, 2005.].

A base para estas discussões é que, ao invés de continuar a busca para encontrar um processo de software perfeito os pesquisadores precisam reconhecer que cada organização precisa desenvolver suas próprias práticas de auto-aperfeiçoamento, e estabelecer uma infraestrutura de melhoria apropriada [Birk & Rombach, 2001]. Uma abordagem ágil para MPS seria aquela que fosse flexível às necessidades locais, que possibilitasse estimular a inovação no processo, construir a melhoria em torno daqueles que estão motivados, incentivar a auto-organização de equipes competentes e promover o desenvolvimento sustentável dos processos [Allison, 2005].

Uma das abordagens para MPS ágil é sugerida por Salo (2007) que enfatiza que enquanto os métodos tradicionais de MPS geralmente focam na melhoria contínua dos processos, os princípios do MPS ágil devem ser iterativos, adaptativos e prover o aperfeiçoamento das atividades dos times de

desenvolvimento de software para aumentar a eficácia [Salo, 2007]. O desenvolvimento ágil de software fornece uma abordagem pouco tradicional para MPS já que o conhecimento e experiência dos desenvolvedores de software são reconhecidos e valorizados [Salo, 2007]. O MPS ágil deve ser apto a acolher as mudanças ao longo de um projeto, sejam estas de requisitos, sejam por aspectos técnicos. Este é o ponto onde ajustes são feitos no processo a partir de práticas, métodos, e ferramentas para responder às mudanças, quando necessário. Isso explica a preferência por abordagens indutivas a prescritivas.

Outra abordagem de MPS ágil é sugerida por Moe (2011), que mudou o foco de apoio às atividades de gestão de projetos para apoiar todo o time. O desenvolvimento dirigido a mudanças proporciona uma forte infraestrutura para MPS. Esta forma de MPS se caracteriza por apoiar o time inteiro, não só a gestão do projeto, e é desejado elevar o nível das práticas a partir de ciclos curtos de aprendizagem. Ou seja, é mais valioso melhorar a maneira como o time executa as práticas do que melhorar o processo em si.

**Discussão da Seção:** A definição de MPS Ágil utilizada em toda esta pesquisa pode ser assim expressa: **MPS ágil é um conjunto de práticas adotadas pelas organizações com o objetivo de torná-las mais adaptáveis.** Assim sendo, qualquer iniciativa de MPS ágil tem o intuito de tornar as organizações mais aptas para responder a mudanças.

MPS ágil possui a característica de ser indutivo, não seguindo nenhum modelo ou norma para a melhoria. A organização deve identificar onde é preciso melhorar e identificar também a forma como o processo deve ser melhorado. Normalmente isso é baseado na experiência do próprio pessoal.

Podemos perceber que as abordagens mais usadas são: (i) a melhoria do comportamento do time (Salo, 2007), e (ii) a melhoria das práticas (Moe, 2011). A abordagem que propõe a melhoria dos times é a que os times ágeis utilizam com mais frequência. Ela é vista em XP a partir do princípio de reflexão [Beck, 2004], nas retrospectivas do Scrum [Kerth, 2001] e Kaizen [Shore e Warden, 2008]. A abordagem proposta para melhoria de práticas [Moe, 2011] não é comumente encontrada nos relatos da comunidade ágil. Estes conceitos são

novidades encontradas nas ciências administrativas, tais como os sistemas complexos adaptativos [Vidgen e Wang, 2006] e Inovação Aberta [Oza e Abrahansson, 2009]. A proposta de ambas as abordagens é deixar a organização o mais preparada possível para mudanças em ambientes incertos, e todas as mudanças propostas são realizadas com base nos objetivos de negócio da organização.

Existe outro conceito de melhoria de processo de software que é utilizado em empresas ágeis, que é baseado no enxugamento do processo até que mais nada possa ser retirado. Este conceito está associado ao desenvolvimento enxuto de software ou *Lean Software Development* [Poppendieck e Poppendieck, 2003]. O principal conceito de melhoria envolvida nesta abordagem enxuta se chama mapeamento de cadeia de valor ou *value stream mapping* [Poppendieck e Poppendieck, 2003]. Nele, a organização deve avaliar em seu processo o que cada etapa gera de valor e eliminar aquelas cujo custo é maior do que o valor agregado. Embora esta técnica seja muito utilizada em organizações ágeis não podemos considerá-la, conceitualmente, como ágil por dois motivos:

(1) Na indústria de manufatura onde nasceram os conceitos enxutos e ágeis, o nascimento do termo ágil [Iacocca, 1991] implicava numa evolução do termo enxuto. Lá o termo ágil significava aproveitar a leveza promovida pela abordagem enxuta para preparar-se para mudanças.

(2) Na própria reunião do fechamento do acordo ágil foi determinado que as metodologias leves (*lights*) não iriam continuar com este nome. Pois, o foco das metodologias ágeis era estar apto a mudanças, e não eliminar a documentação.

Assim, podemos concluir que o mapeamento da cadeia de valor não pode ser considerado uma abordagem ágil para MPS, uma vez que sua principal motivação está em eliminar o desperdício, e não abraçar a mudanças. É importante ressaltar que os dois principais valores das abordagens enxutas são: (i) melhoria contínua e (ii) respeito às pessoas. Entretanto, este conceito pode ser usado também em ambientes de otimização (não ágeis), e isto também é verdade

para o *value stream mapping*.

Vale ressaltar que métodos ágeis possuem um funcionamento sinérgico com as abordagens enxutas. Isso significa que métodos ágeis tem um maior sucesso quando trabalham em conjunto com abordagens enxutas, embora tratem de conceitos diferentes que tratam de problemas diferentes [Wang e Conboy, 2011].

Esta seção é importante para fechar o entendimento da discrepância existente entre o MPS ágil e o MPS baseado em modelos de maturidade, já que os últimos são baseados na repetição de um processo ãpreferencialmenteö estático (otimização) enquanto os primeiros estão adequando o processo para que mudanças sejam tratadas de forma adequada. As diferenças existentes entre as duas abordagens de MPS será apresentada na Tabela 10 a seguir.

**Tabela 10. Diferenças entre as abordagens de MPS Tradicional e Ágil.**

<b>Características</b>	<b>MPS Tradicional</b>	<b>MPS Ágil</b>
<b>Base para a melhoria</b>	<i>Guias de recomendação contendo as boas práticas.</i>	<i>Experiência das pessoas que sugerem as mudanças.</i>
<b>Implementação da Melhoria</b>	<i>Realizada a partir de projetos de melhoria.</i>	<i>Realizada pela equipe dentro dos projetos.</i>
<b>Avaliação da melhoria</b>	<i>Baseada em critérios objetivos e de preferência quantitativos.</i>	<i>Baseado na avaliação subjetiva da equipe.</i>
<b>Objetivo do MPS</b>	<i>Otimização. Produzir mais, melhor e mais barato.</i>	<i>Adaptação. Resposta rápidas a mudanças.</i>
<b>Definição do Processo</b>	<i>O Processo é definido e desenhado para se tornar o processo base para melhoria.</i>	<i>Não é definido com detalhes. Pois, mudanças no ambiente podem requerer mudanças radicais no processo.</i>
<b>Validação e Verificação</b>	<i>Normalmente realizadas por pessoas externas a equipe em processos de auditoria.</i>	<i>Realizada pela própria equipe, com o apoio de ferramentas durante o projeto.</i>

## 2.5 EROÇÃO DE PROCESSO

Humphrey (1989) define a erosão de processo como o fenômeno que ocorre

quando, em uma organização, o processo atual é ignorado ou não utilizado, e a equipe passa a utilizar práticas *ad-hoc* em seu processo atual [Humphrey, 1989]. Segundo Coleman e Connor (2008), as principais causas da erosão do processo são o custo e a adesão dos funcionários a este. A erosão do processo leva eventualmente a um processo mínimo, que é o processo operacional *õde factoõ*. Este processo mínimo costuma ser mantido até que um evento no nível de negócios exija que ele seja alterado.

Quando ocorre uma iniciativa de MPS dentro de uma organização, surge um novo processo de desenvolvimento de software. Após o período de instalação deste, é iniciado um processo de erosão que retroage em direção ao processo mínimo, que é aquele que mal consegue satisfazer os objetivos de negócio da organização [Basri e O'Connor, 2010]. Alguns dos ganhos obtidos durante a iniciativa de MPS são perdidos, e a organização então se move para um estado de inércia em seu processo. Esta inércia será mantida em relação a qualquer mudança posterior até que outro evento de negócios faça com que um novo ciclo de MPS se repita [Coleman e Connor, 2007].

Isso ocorre porque as empresas experimentam dificuldades em institucionalizar os ganhos do MPS, e a consequência é uma retração que se reflete na clara erosão que se dá imediatamente após a iniciativa de MPS terminar [Coleman, 2006].

A erosão de processo pode ter inúmeras razões, entretanto, a forma mais observada nesta retração ocorre quando um processo novo é estabelecido inicialmente e adaptado de acordo com as exigências locais. Este processo pode ser melhor talvez para atender projetos maiores, mas quando se volta para os projetos menores, muitas vezes algumas etapas do processo são omitidas ou deixadas de lado [Coleman e Connor, 2007]. Em muitos casos, o tamanho do projeto é o fator determinante para a utilização ou não do processo. No entanto, quando se volta a ter um projeto maior, as boas práticas estabelecidas muitas vezes não são reintroduzidas de volta ao processo [Coleman e Connor, 2007].

É importante notar que existe uma diferença entre a adaptação de processo e processo mínimo. A adaptação é consciente e de esforço deliberado

para executar uma ação de modelo genérico que tenha em conta o contexto local. O processo mínimo obtido a partir erosão do mesmo representa uma redução adicional desse procedimento já adaptado [Coleman, 2006].

**Discussão da Seção:** A erosão de processo é um fenômeno comum quando uma organização toma a iniciativa de modificar seus processos sem que aqueles que o executam entendam o porquê de cada elemento do mesmo. Quando isso ocorre, existe a tendência de que o processo comece a ser executado de forma *ad-hoc*, ou de forma mínima para que as atividades básicas sejam executadas.

Embora não haja pesquisas nem dados quantitativos sobre a erosão do processo em empresas brasileiras, podemos inferir que isso ocorra também nas empresas que se submetem a uma iniciativa de MPS baseada em modelos de maturidade. Isso é sugerido pelas empresas MPS.BR que não se re-certificaram (1/3 delas não fizeram isso). Outro sintoma é dado pelo próprio relatório iMPS [Travassos e Kalinowski, 2011] que mostra que uma parcela significativa das empresas perde produtividade, aumenta os custos e prazos, e metade delas sequer obteve o retorno investido no programa de MPS.

Isso sugere que as empresas não tiveram tempo de se beneficiar com os ganhos do programa e voltaram a adotar algo mais próximo do seu processo mínimo. Vale ressaltar que este último ponto é especulativo do ponto de vista de Engenharia de Software Experimental, mas na prática é comum encontrar empresas que abandonaram algumas práticas após a avaliação nestes modelos. Este tópico, erosão de processo, se mostra importante para esta tese devido a ser considerado o primeiro sintoma de que as empresas que tomaram a decisão de se certificar não estão estruturadas para manter aquele processo, ou não é interessante para elas do ponto de vista de negócios. Esta tese vai apresentar uma evolução dessa erosão em algumas empresas questionadas.

## 2.6 TRABALHOS RELACIONADOS

Observando a literatura sobre a relação entre métodos ágeis e modelos de

quando, podemos encontrar diversos trabalhos que retratam este relacionamento sobre os mais diversos aspectos. A seguir, os trabalhos relacionados são apresentados por categoria.

O primeiro grupo de trabalhos relacionados apresentados são as resenhas e reflexões teóricas sobre a relação entre o CMMI e os métodos ágeis, bem como na hibridização destas abordagens. Os trabalhos listados são: [Glazer 2001; Boehm, 2002; Boehm & Turner, 2003; Turner, 2002; Turner & Jain, 2002; Theunissen et al., 2003; Cohen et al. 2004; Anderson, 2005; Konrad & Over, 2005; Hurtado & Bastarrica, 2006; Vinekar et al., 2006; Fritzsche & Keil, 2007; Glazer et al., 2008; Boehm & Turner, 2010]. Estes trabalhos apresentam duas linhas de pensamento distintas: (1) de que as abordagens são antagônicas ou (2) que as abordagens podem coexistir no mesmo ambiente.

Para a linha de pensamento (1), a principal divergência é a quantidade de atividades existentes nos processos tradicionais que não geram software. O excesso de documentação e as atividades de controle tornam o processo pesado e com uma quantidade de pessoas maior do que o necessário para a sua realização. Então, o foco do desenvolvimento de software deveria deixar de serem os processos, as ferramentas, a documentação excessiva, o controle rígido e o planejamento extenso, para ser a entrega de software da forma mais rápida possível.

É a partir destes trabalhos que surgiu o senso comum de que modelos de maturidade são pesados e burocráticos enquanto os métodos ágeis são desprovidos de documentação, planejamento e controle sendo, confundidos com a programação *cowboy* [Highsmith e Cokburn, 2001].

A linha de pensamento (2) prega que as duas abordagens possuem alguma sinergia e que o CMMI é um modelo, enquanto os métodos ágeis são métodos, e assim há a clara separação entre o *quê* fazer (CMMI) e *como* fazer (XP e Scrum). Esta visão é a mais aceita hoje para quem utiliza processos ditos híbridos, ou seja, aqueles que utilizam métodos ágeis e modelos de qualidade no mesmo processo.

**Relevância para a Tese:** Esta tese pouco se baseou nestes trabalhos

teóricos, porque os objetivos destes trabalhos são: (1) apontar os problemas do CMMI e do peso colocado em cima do ciclo de vida cascata, pelo qual o CMMI é estigmatizado ou (2) apontar a divisão das responsabilidades ao separar o que fazer (modelos de qualidade) de como fazer (métodos ágeis), e a partir dessa separação idealizar uma coexistência harmônica entre eles.

Nenhum dos trabalhos acima levou esta discussão para o nível de melhoria de processo, nem sobre como a mesma deveria ocorrer, ou que tipo de diferenças existem entre as duas abordagens de MPS.

O segundo grupo de trabalhos que envolvem as abordagens ágeis e os modelos de qualidade são os mapeamentos entre o CMM, CMMI e MPS.BR em relação a abordagens ágeis tais como XP e Scrum. Os trabalhos nesta categoria são: [Paulk, 2001; Nawrocky et al., 2002; Martinsson, 2003; Reifer, 2003; Vriens, 2003; Bos & Vriens, 2004; Kähkönen & Abrahamsson, 2004; Santana et. al, 2006; Marçal et al., 2007; Marçal et al., 2008]. Estes mapeamentos são trabalhos que exemplificam a relação *versus* como e propõem realizações concretas da junção dos modelos de qualidade e métodos ágeis.

Nestes mapeamentos as práticas específicas do CMMI e resultados esperados do MPS.BR foram mapeados para práticas dos métodos Scrum e XP, que neles o modelo de maturidade indicava o que fazer enquanto o método ágil provia o como fazer. Estes mapeamentos representam pontos de vista teóricos sobre como a junção do CMMI/MPS.BR com métodos ágeis poderia ser promovida dentro de uma organização. Estas pesquisas propunham o *como* realizar as atividades que o CMMI/MPS.BR exige, com base em métodos ágeis.

**Relevância para a Tese:** Existem dois pontos relevantes que devemos considerar sobre estes mapeamentos:

(1) Que estes não estão preocupados em como irá ocorrer a melhoria do processo de software dentro da organização, e sim em como utilizar os métodos ágeis como um repositório de práticas para se atingir os níveis de maturidade do CMMI e MPS.BR. Esta visão expõe uma ausência de análise da utilização de metodologias ágeis no negócio da organização e isso leva ao uso dos métodos ágeis como ferramenta de implementação do CMMI e/ou MPS.BR.

A incoerência que essa visão pode trazer é a utilização direta das abordagens ágeis como Scrum e XP, no quarto nível do acordo ágil, sem compreender os valores e princípios norteadores (níveis mais altos da pirâmide) dos métodos ágeis e em que contexto eles devem ser aplicados.

A falta do entendimento do contexto da utilização dos métodos ágeis é apontada em casos como uma das razões de falha na adoção destes. Esse fato já era discutido por Keenan (2004) antes da sua popularização, e foi ratificado por Krutchen (2011) recentemente.

(2) A outra razão é que essa visão *õferramentalõ* das abordagens ágeis dentro dos modelos de maturidade leva os próprios mapeamentos a considerar apenas as práticas específicas (CMMI) e resultados esperados (MPS.BR), que são elementos pontuais nos modelos, e desconsiderar as práticas genéricas (CMMI) e atributos de processo (MPS.BR), que são elementos norteadores da organização. O que acontece com estes mapeamentos é que eles propõem sugestões pontuais sobre onde e como usar métodos ágeis dentro de ambientes CMMI/MPS.BR (Elementos Pontuais). No entanto, esquecem-se de integrar os métodos ágeis dentro da política da organização (Elementos Norteadores).

Devemos levar em consideração que esses mapeamentos ainda apresentam muitas lacunas entre as abordagens ágeis e os modelos de maturidade, então é pertinente considerar pequena a parcela de agilidade sugerida por esses mapeamentos dentro do contexto do CMMI.

O que esta tese propõe é retirar o caráter *õferramentalõ* dos métodos ágeis nesta relação de *õcomo fazerõ versus õo que fazerõ*. Levando para o assunto melhoria de processo de software, esta pesquisa tenta avaliar organizações que a iniciativa de melhoria de processo de software fosse baseada em métodos ágeis e assim tornar o processo criado pela organização mais apto a mudanças do que aqueles que apenas considerem os métodos ágeis como ferramentas dentro do seu processo.

O terceiro grupo de trabalhos relacionados são relatos de experiências bem sucedidas do desenvolvimento ágil em ambientes CMMI ou MPS.BR. Os trabalhos aqui relacionados são: [Lycett et al., 2003; Baker, 2005; Baker, 2006;

Onveira et al., 2007; Sutherland et al., 2007; Jakobsen & Johnson, 2008; Santana et al., 2008; Vriens & Barto, 2008; Jakobsen & Sutherland, 2009; Santana et al., 2009a; Santana et al., 2009b; Catunda et al., 2010; McMahon, 2010; Trujillo et al., 2010; Catunda et al., 2011; Kovacheva & Todorov, 2011; Ronkko et al., 2011].

Estes trabalhos apresentam casos reais de empresas que utilizaram métodos ágeis em ambientes CMMI ou MPS.BR, e todos reportam lições aprendidas durante a utilização conjunta das abordagens, apresentando soluções mais práticas do que aquelas providas nos mapeamentos.

Apenas um trabalho [Jakobsen e Sutherland, 2009] apresenta resultados quantitativos sobre a junção das duas abordagens no mesmo processo. Essa avaliação foi considerada positiva com ROI de 300% em alguns casos.

**Relevância para a Tese:** A relevância destes trabalhos para esta tese é a mesma dos trabalhos do grupo anterior (Mapeamentos). Estes trabalhos mostram na prática como os mapeamentos foram realizados. As contribuições trazidas por estes casos de vida real são as lições aprendidas e os resultados quantitativos da utilização de tais abordagens. Contudo, continuamos com a visão ôferramentalõ do uso das metodologias ágeis dentro dos modelos de maturidade de processo. Nenhum destes artigos considera qualquer forma de melhoria de processo de software, nem a baseada em modelos de maturidade, nem a abordagem de MPS ágil.

O quarto grupo de trabalhos é aquele que trata da melhoria de processo de software relacionado às abordagens ágeis. Para este grupo os trabalhos encontrados são: [Salo, 2007; Pino, 2010; Moe, 2011; Fægri, 2012].

Salo(2007)sugereque a melhoria de processo de software ocorrida nos times ágeis é percebida na melhoria de comportamento deste time para ficar mais eficiente e efetivo. Isto é realizado a partir das reuniões de retrospectivas após cada iteração. Esta forma de melhoria de processo se apresentou como revolucionária se comparada ao pensamento tradicional (prescritivo) de MPS e trouxe à luz novas perspectivas para a melhoria de processo de software.

Moe (2011) comenta que um fator que auxilia a MPS, em ambientes

orientados a mudanças, e o total suporte ao autogerenciamento para que as equipes possam levar esta melhoria do nível de time proposta por Salo (2007) para as práticas do dia a dia, e que isso é refletido diretamente na melhoria das práticas realizadas pelas equipes no dia a dia.

Fægri (2012) apenas aponta que existem novas tendências para a melhoria de processo de software quando se trata de métodos ágeis.

Pinoe Salgado (2010) apresentaram casos reais onde houve a implementação de um programa de MPS ágil dentro das organizações. Salgado (2010) foi mais sucinto, apenas apresentando como implantar processos na organização de forma ágil. Não considerou outros elementos como o modelo de MPS, métodos de avaliação, ou sequer adaptação de processos.

Pino (2010) propôs um modelo de MPS ágil que poderia ser complementar ao processo de Salgado (2010), que só tratava da implantação. Pino apresentou boas práticas e conceitos de como fazer o MPS ágil de forma iterativa e incremental.

**Relevância para a Tese:** A relevância dos trabalhos de Salo (2007), Moe (2011) e Fægri (2012) é pequena visto que eles apresentam sugestões sobre como ocorre MPS em abordagens ágeis. E na verdade não é a nossa preocupação relatar o como fazer o MPS ágil, e sim avaliar a eficácia dessas abordagens que não necessariamente foram utilizadas nas empresas do escopo deste trabalho. Já os trabalhos de Pino (2010) e Salgado (2010) são mais importantes por que este é o tipo de MPS Ágil que foi encontrado na realização da pesquisa, que é iniciar o programa de MPS baseado na adoção de métodos ágeis.

A grande importância do trabalho do Pino (2010) é que ele mostra que existe uma diferença significativa em como a melhoria organizacional ocorre entre as empresas que utilizam a o MPS tradicional e o MPS ágil.

O último grupo identificado foram os trabalhos que envolvem os métodos ágeis, CMMI e Melhoria de Processo de Software. No primeiro trabalho, Pikkarainen e Matyniemi (2006) propõem uma maneira de avaliar a utilização de metodologias ágeis em ambientes CMMI. Eles também propuseram que a

o menor impacto da utilização das metodologias ágeis em uma determinada organização também seria observada na avaliação pelo CMMI.

No segundo trabalho, Pikkarainen (2008) propõe a melhoria de processo de software baseada em modelos de MPS com o auxílio dos métodos ágeis. Foi apresentado um caso prático onde uma organização CMMI avaliou sua melhoria integrando os métodos ágeis dentro de seu processo, e avaliando a utilização da parte ágil baseada no CMMI.

**Relevância para a Tese:** Apesar dos trabalhos envolverem as mesmas palavras chave (Métodos Ágeis, MPS e Modelos de Maturidade), eles quase não auxiliam no desenvolvimento desta tese, nem proveem resultados comparativos que possam ser usados no contexto deste trabalho. Isto é percebido pelo foco dado no CMMI e na execução/avaliação de sua forma de melhoria de processo (Prescritiva), e os métodos ágeis continuam sendo usados como ferramentas na implementação do CMMI. Na prática essas pesquisas apenas mudam o foco da pesquisa, que era dos mapeamentos propostos dos métodos ágeis para os modelos de maturidade para como a melhoria de processo (prescritiva) ocorre neste mesmo ambiente.

## 2.7 RESUMO DO CAPÍTULO

A literatura que trata sobre modelos de maturidade e os métodos ágeis não evoluiu muito ao longo dos últimos 11 anos. A grande maioria dos trabalhos se coloca favorável à relação *versus* proposta pelos primeiros trabalhos publicados sobre este relacionamento. Assim, a única evolução perceptível foi dos trabalhos iniciais que apontavam as diferenças teóricas entre as abordagens. Isso evoluiu para os mapeamentos teóricos onde abordagens como Scrum e XP eram utilizadas para implementar os modelos de maturidade. Essas pesquisas evoluíram para a prática, e os casos de vida real e vários trabalhos relatam a adoção de métodos ágeis nestes ambientes e, por último, a avaliação de como ocorre o MPS em ambientes CMMI que utilizam práticas ágeis.

Entretanto, os métodos ágeis apresentam suas próprias necessidades de MPS, isso é levantado por Salo (2007) e ratificado por Moe (2011) que são diferentes daquelas apresentadas nas abordagens tradicionais. A partir deste conceito, surgiram as primeiras formas de conduzir programas de MPS e algumas destas baseadas em métodos ágeis [Pino, 2010; Oliveira, 2010].

O objetivo desta tese é avaliar a implementação dos modelos de maturidade CMMI e MPS.BR a partir de um programa de MPS que seja baseado em métodos ágeis. Diante de tudo que foi analisado nesta seção não existem trabalhos que apontem uma iniciativa de MPS baseada em métodos ágeis para organizações que procuram os modelos de maturidade, muito menos trabalhos que os avaliem.

### 3. Metodologia de Pesquisa

Este capítulo apresenta a metodologia de pesquisa adotada nesta tese.

#### 3.1. ABORDAGEM DE PESQUISA

Os estudos de caso remetem à tradição hermenêutica de buscar o entendimento "visto por dentro" [Evered e Lous, 1981]. O estudo de caso faz isso a partir da aproximação com a organização estudada e deve-se buscar a compreensão mais completa possível daquele contexto, explorando uma infinidade de fontes de dados já que este método investiga um fenômeno dentro do seu contexto de vida real [Yin, 2002].

O método de estudo de caso pode ser útil para várias finalidades, por exemplo, [Yin, 2002]:

- Explicar ligações causais em intervenções da vida real que são complexas demais para a estratégia de pesquisa experimental;
- Descrever uma intervenção e seu contexto na vida real;
- Ilustrar determinados tópicos dentro de uma avaliação;
- Explorar as situações em que a intervenção não tem conjunto claro e único dos resultados, e;
- Realizar uma avaliação da meta, isto é, um estudo de um estudo.

Geralmente, o estudo de caso é importante porque a pesquisa exige uma produção sistemática de suas evidências [Flyvbjerg 2001].

Nesta pesquisa, foi identificado que algumas empresas do mundo real estariam interessadas em adotar uma estratégia de melhoria de processo de software baseada em métodos ágeis, logo, avaliar o desempenho em cada organização se mostra um estudo a ser conduzido por um estudo de caso.

O início desta pesquisa se deu em 2008, precisamente, quando Santana (2008) identificou que existiam alguns problemas ao utilizar os métodos ágeis na obtenção de certificações no nível 2 do modelo CMMI. Nos últimos três anos foram iniciados os estudos de caso para avaliar os indicadores apresentados por

empresas que utilizaram essa forma de MPS.

Esta pesquisa é interpretativa, o que significa que nela existe uma análise realizada pelo pesquisador para o significado atribuído às observações. O pesquisador define como serão realizadas as observações empíricas, que consistem em passagens descritivas, bem como analíticas. A intenção é fornecer ao leitor uma sensação de "mundo real" [Strauss & Corbin, 1998].

O estudo de caso compreende a abordagem de pesquisa escolhida para esta tese, e em complemento ao estudo de caso também foi utilizado o método de pesquisa *survey* [Pfleeger e Kitchenham, 2001]. A análise de dados será quantitativa e comparativa para as QP1 e QP2. Já na última questão de pesquisa (QP3) será utilizada outra abordagem para análise que é a teoria fundamentada em dados [Perry et al. 2000].

Segundo elas um *survey* não é apenas o questionário para a coleta de informações, é um sistema completo de coleta de informações para descrever, comparar ou explicar conhecimentos, atitudes, e comportamento. Assim, o instrumento de pesquisa é parte de um maior processo de pesquisa com atividades claramente definidas que são [Pfleeger e Kitchenham, 2001]:

- Definir objetivos específicos e mensuráveis;
- Planejar e programar a pesquisa;
- Garantir que os recursos adequados estejam disponíveis;
- Projetar a pesquisa;
- Preparar o instrumento de coleta de dados;
- Validar o instrumento de pesquisa;
- Selecionar participantes;
- Administrar e pontuar o instrumento;
- Analisar os dados;
- Comunicar os resultados.

Muitas das fases do método *survey* proposto por Pfleeger e Kitchenham (2001) sobrepõem as fases do estudo de caso propostas por Brereton et al. (2008) e por isso serão consideradas a mesma fase.

Para questão QP3, um questionário aberto foi utilizado e sua análise

quantitativa de dados foi guiada pelos princípios da teoria fundamentada, como descrito por Strauss e Corbin (1998). A teoria fundamentada nos dados é um método de pesquisa que procura desenvolver uma teoria baseada em dados, que foram sistematicamente recolhidos e analisados.

Segundo Strauss e Corbin (1998) a teoria fundamentada é aquela que é indutivamente derivada do estudo do fenômeno que ela representa. Ou seja, ela é descoberta, desenvolvida e, provisoriamente, verificada através da obtenção sistemática de dados e análise de dados relativos a esse fenômeno. Portanto, a coleta de dados, análise, e teoria estão em relação recíproca umas com as outras. A teoria fundamentada não começa com uma teoria para que depois a mesma seja provada, pelo contrário, começa com uma área de estudo que é relevante para essa pesquisa e a teoria surge a partir das avaliações dos dados.

Vários materiais que descrevem um evento ou um fenômeno foram coletados no questionário. Isto permitiu o levantamento de uma variedade de estratégias para analisar o material [Langley 1999]. Primeiro, o contexto foi descritoem uma narrativa para compreender o que ocorreu na empresa estudada. Em seguida, a análise foi realizada em todas as fontes de dados e os resultados sintetizados.

O quadro metodológico desta pesquisa é apresentado na Tabela 11.

**Tabela 11 - Quadro Metodológico**

<i>Quadro Metodológico</i>	
<i>Abordagem de Pesquisa</i>	<i>Dedutiva</i>
<i>Natureza das Variáveis</i>	<i>Qualitativas/Quantitativas</i>
<i>Método de Pesquisa</i>	<i>Estudo de Caso</i>
<i>Objeto</i>	<i>Interpretativo</i>
<i>Escopo</i>	<i>Pesquisa de Campo</i>

### 3.2. CONSTRUÇÃO DA PESQUISA

Brereton et al.(2008)definem sete passos para a condução de um estudo de caso. Estes passos, mostrados na Figura 9 a seguir, serão detalhados nesta seção:

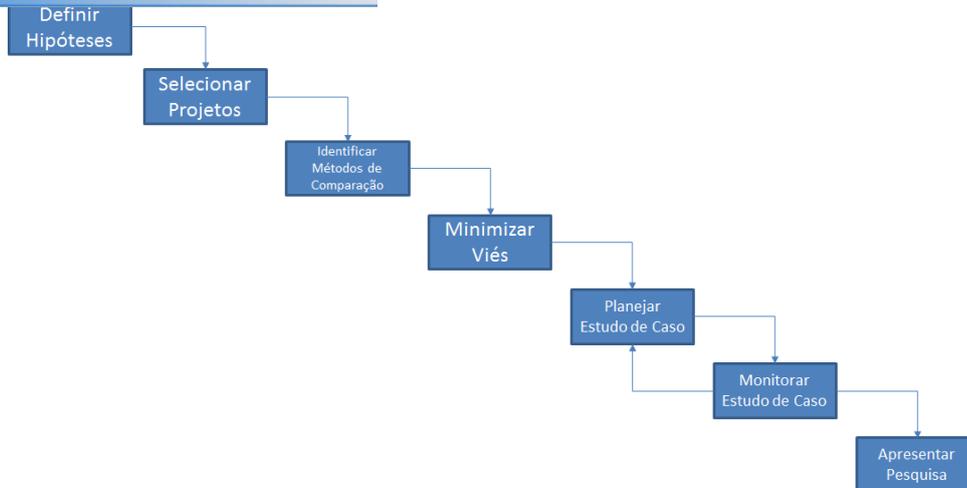


Figura 9 – Passos para Realização de um Estudo de Caso [Brereton et al, 2008]

### *Passo 1. Definir a Hipótese.*

Esta etapa foi baseada nas lacunas de pesquisa e nas oportunidades observadas nos estudos de casos de vida real. Nesta pesquisa, esta etapa foi concebida inicialmente com a definição de três questões de pesquisa:

*QP1 ó É possível executar programas de melhoria de processo de software baseados em metodologias ágeis com o objetivo de obter a certificação CMMI nível dois e MPS.BR nível F?*

**Objetivo:** A resposta desta questão de pesquisa será um ôbooleano (Se certificou ou não se certificou) indicando o fato da obtenção do nível CMMI com esta nova forma de MPS. Uma resposta negativa tornaria nulas as próximas duas questões de pesquisa.

Serão tecidos comentários sobre esta questão de pesquisa baseados nos questionários abertos que foram enviados para coleta dos dados da QP3. Essa avaliação ôbooleana será feita observando os resultados de avaliações divulgados pelo Sistema de Avaliação de Resultados Publicados do CMMI<sup>5</sup>, e da página oficial de avaliação do MPS.BR<sup>6</sup>. Será conferido se as empresas

<sup>5</sup> <https://sas.sei.cmu.edu/pars/pars.aspx>

<sup>6</sup> [http://www.softex.br/mpsbr/\\_avaliacoes/default.asp](http://www.softex.br/mpsbr/_avaliacoes/default.asp)

realmente constam nestes meios oficiais como aprovadas em avaliação.

*QP2 ó Quais os benefícios observados por estas organizações ao se adotar a melhoria de processo baseada em métodos ágeis? E em longo prazo?*

**Objetivo:** A resposta desta questão de pesquisa é obtida por dados quantitativos baseado nos indicadores de desempenho das empresas. Estes indicadores devem mostrar a evolução da organização, seu desempenho antes de iniciar o programa *versus* depois da avaliação, bem como ser comparados com os resultados do iMPS [Travassos e Kalinowski, 2011] para avaliar seu desempenho em relação ao total de empresas avaliadas no MPS.BR.

No primeiro levantamento respondido pelas empresas (Apêndice B) são providas informações sobre quando foi iniciado o programa de melhoria. Se a empresa sabe o seu faturamento e produtividade atuais, se a mesma sabe a duração e custo de seus projetos para que sejam comparados no segundo e terceiro levantamentos (Apêndice C e D respectivamente), e se esses indicadores de desempenho estão melhores do que quando as mesmas não eram certificadas. No caso do terceiro levantamento, os dados coletados também são comparados com aqueles levantados na última coleta para verificar a erosão do processo.

Desta forma foi possível montar um gráfico comparativo com os resultados do iMPS.

*QP3 ó Como a melhoria de processo ágil ãIndutivaö está relacionada à melhoria de processo ãPrescritivaö adotada pelo CMMI e MPS.BR?:*

**Objetivo:** A resposta desta questão será obtida a partir de questionários enviados às empresas do estudo de caso. Para esta questão de pesquisa foi escolhido o método *survey* [Kitchenham, Pickard e Pfleeger, 1995] e a análise baseada na teoria fundamentada em dados [Strauss e Corbin, 1998]. Aqui devemos captar que características ágeis são encontradas no MPS dessas organizações.

Nesta parte, quatro questões abertas foram colocadas no terceiro levantamento (Apêndice D), e elas foram avaliadas para responder esta pergunta de forma qualitativa. Também são buscadas informações qualitativas de

empresas que não conseguiram finalizar o programa com sucesso. As perguntas realizadas a estas empresas estão no formulário presente no Apêndice E.

Esta fase do estudo de caso coincide com a fase de Definir objetivos específicos e mensuráveis do *survey*.

### *Passo 2. Selecionar os Projetos Piloto.*

O perfil de projeto desejado para participar deste estudo de caso envolve empresas que pretendem obter as duas certificações (CMMI e MPS.BR) e que deveriam iniciar o programa de melhoria de processo de software baseado em abordagens ágeis. Ao todo 12 empresas identificadas se encaixariam neste critério, entretanto, uma delas não respondeu o questionário e três foram descartadas por que só adotaram métodos ágeis numa fase tardia do programa de MPS, e este era um dos critérios de exclusão. Ao todo oito empresas fizeram parte de todo o estudo. Destas, seis conseguiram as certificações CMMI e MPS.BR enquanto duas abortaram o processo.

Vale ressaltar que dessas oito organizações que tiveram suas informações mantidas, duas delas não responderam a nenhum dos questionários, mas essas não foram descartadas do processo, pois existiam evidências de que se encaixariam no perfil desejado, contribuindo para a resposta da QP1.

Não se sabe exatamente o número total de empresas que possuem as duas certificações e iniciaram seus programas de MPS baseado em métodos ágeis. A única indicação do tamanho desta amostra é que apenas 42 empresas possuem níveis CMMI e MPS.BR (População máxima), mas ainda precisamos retirar aquelas que não adotaram abordagens ágeis ou as que usaram tardiamente nas atividades do MPS.

12 empresas com este perfil foram encontradas a partir do método conhecido como *amostragem bola de neve* [Kitchenham e Pfleeger, 2002d]. Esta abordagem trata de pedir às pessoas que participaram de uma pesquisa para nomear outras pessoas que elas acreditam que estariam dispostas a participar, ou que se encaixem no perfil desejado. A busca por uma maior amostragem

continua até que o número necessário de respostas seja obtido ou até que ninguém mais possa ser indicado. Esta técnica é frequentemente usada quando é difícil a identificação da população.

Para todas as 12 empresas foi enviado um e-mail (Apêndice A) convidando para a pesquisa. Dessas, 11 responderam, o que significa uma taxa de resposta de 91,5%. Das 11, duas responderam que não gostariam de continuar a colaborar com a pesquisa, embora tenham respondido os outros dados do primeiro questionário, e três delas iniciaram a adoção de métodos ágeis muito tempo depois do início do programa de MPS.

Das seis restantes, duas não conseguiram obter as certificações de qualidade e responderam os questionários do Apêndice B e E. Outras três empresas responderam aos questionários do Apêndice B, C e D, enquanto uma respondeu aos questionários B e C.

Isso significa que quatro empresas que se certificaram no CMMI e MPS.BR e iniciaram seus trabalhos utilizando abordagens ágeis seguiram na pesquisa, e isso implica numa amostra de cerca de 10% da população máxima possível. Acreditamos que esse número represente uma parcela maior da população de empresas que possuem as duas, já que as empresas adotaram as metodologias ágeis depois de iniciar seu programa de MPS ou o programa de MPS não tinham a ideia da agilidade.

Depois de identificadas as empresas que estariam aptas a participar da pesquisa, foi detectado nelas quem eram as pessoas que fariam parte da unidade de amostra. Essas pessoas eram gerentes ou desenvolvedores que participaram do programa de MPS da organização e que tinham contato com os projetos ágeis da empresa. Conseqüentemente, poderiam fornecer as informações necessárias para a pesquisa.

Esse contato não foi direto, mas sim mediado por uma pessoa em cada empresa que recolhia todas as opiniões e colocava em um único questionário de resposta.

Todos os cuidados relativos à significância do tamanho da amostra como, qual seria a unidade amostral, que método utilizar para identificar membros

pertencentes a população e que poderiam ser incorporados a amostra, e identificar as taxas de, foram seguidos de acordo com o recomendado por Kitchenham e Pflieger (2002d).

Esta etapa do estudo de caso compreende alguns passos do *survey* que são: (1) Garantir que os recursos adequados estejam disponíveis e (2) Seleção de participantes.

### *Passo 3. Identificar o método de comparação.*

Cada questão de pesquisa possui seu próprio método de comparação. Na QP1 só é necessário identificar os resultados na Página de Publicações dos Resultados do CMMI<sup>7</sup> e do MPS.BR<sup>8</sup> se as empresas foram bem sucedidas ou não nas avaliações oficiais. Este resultado também será comentado de acordo com as respostas dadas nos questionários presentes no Apêndice D e algumas questões qualitativas também serão mostradas, mas muito mais como complemento do que como objetivo de responder a pergunta.

Para responder a QP2 são necessários mais informações, uma vez que é necessário avaliar três dimensões distintas na implementação deste programa de MPS.

- Informações sobre o processo de Implementação do CMMI (Quanto tempo demorou a implantação, quanto custou em relação ao orçamento da empresa, se houve interrupção do programa de MPS).
- Informações sobre o desempenho da organização antes/depois da obtenção dos níveis de maturidade. (A produtividade aumentou/diminuiu, o custo médio por projeto aumentou/diminuiu, o faturamento aumentou/diminuiu, a qualidade do produto final aumentou/diminuiu, o retorno de

---

<sup>7</sup> <https://sas.sei.cmu.edu/pars/pars.aspx>

<sup>8</sup> [http://www.softex.br/mpsbr/\\_avaliacoes/default.asp](http://www.softex.br/mpsbr/_avaliacoes/default.asp)

investimento foi positivo/negativo e a duração dos projetos aumentou/diminuiu).

- Informações sobre a erosão do processo (O processo CMMI/MPS.BR ainda está ativo? Qual seria o resultado da avaliação CMMI/MPS.BR de sua organização se ela acontecesse hoje? E os resultados do erosão no scrum?).

Estas informações serão comparadas com os relatórios do iMPS [Travassos e Kalinowski, 2008 a 2011]. Esta comparação não deve prover um resultado conclusivo uma vez que dados coletados de quatro empresas são, do ponto de vista estatístico, insuficientes para se tirar conclusões. Mas é válido constatar como estarão os indicadores das empresas pesquisadas em relação a população do iMPS para que indícios possam ser apresentados.

Para a RQ3 não teremos nenhuma comparação. Esta fase engloba as seguintes fases do Survey: (1) Preparando o instrumento de coleta de dados e (2) validação do instrumento de pesquisa.

#### *Passo 4. Minimizar o efeito de fatores que causam viés.*

Para este ponto a QP1 se mostra livre desses efeitos já que todo o processo é realizado por avaliações de terceira pessoa, SEI e SOFTEX para CMMI e MPS.BR, respectivamente, que avaliam um único aspecto: se as empresas foram aprovadas em avaliações oficiais após o seu programa de MPS.

O mesmo não se pode apontar para a QP2, pois, nela pode haver viés para as duas primeiras dimensões avaliadas: (i) Avaliação do programa de MPS e (ii) informações sobre o desempenho da organização, uma vez que estamos comparando estas empresas com aquelas do iMPS [Travassos e Kalinowski, 2008 a 2011].

O viés pode ser proveniente dos seguintes fatores:

- Método utilizado não possui caso controle: Por se tratar de um estudo de campo em empresas de vida real, estudando fenômenos de vida real e sendo observadas *in vivo*, não é possível montar

um experimento para responder as questões de pesquisa apropriadas e assim não é possível definir um caso controle. Essa questão foi atenuada com a participação de mais de uma organização, sendo um estudo de caso com múltiplas fontes, o que não garante o sucesso, mas atenua os efeitos de não haver um caso base já que mais fontes provêm mais evidências.

- Método de coleta (questionários) pode se mostrar confuso ou inadequado: Questionários podem sofrer vieses de interpretação de quem responde às suas perguntas, seja por falta de entendimento da pergunta em si, seja por que o respondente não está apto a responder aquelas perguntas, ou por que simplesmente a pergunta está fora de contexto. Para atenuar o problema do entendimento e do contexto dos membros, foram colocadas apenas perguntas do tipo sim e não, e o questionário foi entregue a pessoas que foram indicadas por outros membros da empresa como competentes para responder aquelas perguntas. Para diminuir o viés do contexto da pergunta, foram perguntados aspectos contidos no iMPS.
- Método de comparação: O fato de utilizar o iMPS como fonte de comparação pode trazer o viés de que as populações são ligeiramente diferentes. Entretanto, vale ressaltar que o uso do iMPS foi justamente utilizado para diminuir o viés, já que a sua população é considerável dentro do meio do MPS.BR e não existe outra fonte de informações considerando o conjunto de empresas com o perfil desejado. Neste caso, o iMPS traz o resultado de organizações com características próximas das nossas empresas e o viés trazido pelo CMMI, já que o iMPS ignora esse status, é minimizado pela proximidade entre os modelos, tanto que já há recomendações para tais situações reconhecidas pelo próprio comitê do SOFTEX. [Rocha et al, 2009].

- Método de Análise de dados: A teoria fundamentada em dados tem como característica a interpretação destes por quem realiza a pesquisa. Embora toda a fundamentação seja feita nos dados, outras coisas relevantes podem escapar do pesquisador. Esta é uma característica comum em abordagens qualitativas, mas devemos lembrar que o uso do método e sua codificação é um fator de minimização do viés por este se mostrar mais confiável do que uma avaliação *ad-hoc* feita pelo pesquisador.

#### *Passo 5. Planejar o Estudo de Caso*

Foram rodados seis estudos de casos individuais em cada empresa para que depois haja a aglutinação dos dados. Isso é necessário por que as empresas farão suas implementações do programa de MPS em épocas diferente e com durações diferentes. Sendo assim, foi necessário tratar a coleta de dados de cada empresa em momentos diferentes.

Para cada estudo de caso individual houve quatro fases distintas para a coleta de dados:

*Fase 1* ó Identificação das Empresas Candidatas e Descarte das Empresas que não se adequavam à pesquisa;

*Fase 2* ó Avaliação dos indicadores organizacionais antes da empresa iniciar o programa de MPS;

*Fase 3* ó Avaliação dos indicadores organizacionais imediatamente após a avaliação CMMI/MPS.BR (A que viesse por último);

*Fase 4* -Ciclos Semestrais que aconteceriam após a avaliação para avaliar os indicadores organizacionais e os dados do questionário.

Na Fase 1 as empresas foram identificadas a partir de contatos com instituições implementadoras MPS e fóruns de discussão sobre métodos ágeis. Foram identificadas 12 organizações candidatas e para elas foi enviado um E-mail (Apêndice A) onde 11 delas responderam.

Na Fase 2 foram solicitadas informações a respeito do estágio atual da

organização imediatamente antes do início da implementação do programa de MPS. As informações desejadas foram coletadas via questionário fechado, apresentado no Apêndice B, e solicitava as seguintes informações:

- Média do Nokia Test; [Anexo A] ó Entregue por cinco Empresas( , , , e ).
- Informação sobre se a avaliação de lacunas inicial já havia sido feita e registrada pela organização; ó Entregue por seis Empresas ( , , , , e ).
- Indicação de que a empresa estava ciente de sua produtividade atual; ó Entregue por quatro Empresas ( , , e ).
- Indicação de que a empresa estava ciente do tempo médio gasto por projeto; ó Entregue por cinco Empresas( , , , e ).
- Indicação de que a empresa estava ciente da qualidade atual do seu produto; ó Entregue por quatro Empresas ( , , e ).
- Data inicial da Implementação ó Entregue por seis Empresas ( , , , , e ).

Na Fase 3 foram solicitadas informações a respeito do desempenho da organização logo após a obtenção das certificações. As informações desejadas foram coletadas via questionário fechado apresentado no Apêndice C e pedia as seguintes informações:

- Média do Nokia Test; [Anexo A] ó Entregue por quatro Empresas( , , e ).
- Indicação de que a produtividade da empresa aumentou ou diminuiu; ó Entregue por quatro Empresas ( , , e ).
- Indicação de que o tempo médio gasto por projeto na empresa aumentou ou diminuiu; ó Entregue por quatro Empresas ( , , e ).
- Indicação de que a qualidade final do produto aumentou ou diminuiu; ó Entregue por quatro Empresas ( , , e ).
- Indicação de que o faturamento da empresa aumentou ou

diminuiu; - Entregue por três Empresas ( , e ).

- % da relação entre custo total da implementação do CMMI x Faturamento da Empresa ó Entregue por três Empresas ( , e ).

A Fase 4, aconteceria em ciclos que se repetiriam a cada seis meses. Empresas que foram avaliadas em 2009 e 2010 chegaram a entregar três formulários, enquanto uma foi avaliada em 2011 não conseguiu completar os seis meses para responder tal questionário. As informações desejadas foram coletadas via questionário fechado apresentado no Apêndice D e pedia as seguintes informações:

- Média do Nokia Test; [Anexo A] ó Entregue por três Empresas ( , e ).
- Gráficos da Avaliação do CMMI por área de processo - Entregue por três Empresas ( , e ).
- Indicação de que o ROI foi maior ou menor do que 1; ó Entregue por três Empresas ( , e ).
- Indicação de que o Investimento em MPS aumentou ou diminuiu na empresa; ó Entregue por nenhuma Empresa.
- Indicação de que a produtividade estava maior ou menor que a última medição; ó Entregue por três Empresas ( , e ).
- Indicação de que o tempo médio gasto por projeto na empresa estava maior ou menor que a última medição; ó Entregue por três Empresas ( , e ).
- Indicação de que a qualidade final do produto estava maior ou menor que a última medição; ó Entregue por três Empresas ( , e ).
- Indicação de que o faturamento da empresa aumentou ou diminuiu após a última medição; - Entregue por nenhuma empresa.
- Questionário aberto sobre as percepções de agilidade no atual processo da organização ó Entregue por três Empresas ( , e ).

Para as duas empresas que decidiram não prosseguir com o CMMI, também foi enviado um questionário, Apêndice E, que continham duas perguntas e que foram úteis para responder a QP3. As questões eram:

- Motivo da Desistência ó Entregue por duas empresas ( e ).
- Impressões a Respeito do programa de MPS - Entregue por duas empresas ( e ).

A análise dos dados também foi feita seguindo as fases. Assim que todas as empresa entregaram os dados da Fase 2, eles foram catalogados e guardados para posterior análise. O mesmo foi feito para os dados advindos após o término da Fase 3 para todas as empresas. Vale ressaltar que neste ponto já seria possível responder a QP1 e parte da QP2.

O término do recebimento de questionários estava previsto para fim de março, e mesmo com o pequeno atraso na entrega destes, não houve um impacto negativo no cronograma. Esta fase está casada com as seguintes fases do *survey*: (1) Planejamento e programação da pesquisa e (2) Projetando a pesquisa;

*Passo 6. Monitorar o estudo de caso de acordo com o plano.*

Uma vez que o estudo de caso consistiu apenas em coletas de questionários em datas pré-determinadas, não houve grandes desvios no planejamento, já que, devido ao longo tempo existente entre as fases do estudo de caso, havia tempo suficiente para realizar as análises parciais do projeto.

O que estava fora do planejamento foi a avaliação do questionário das empresas ãdesistentesõ, mas houve contribuições importantes para o trabalho através desses questionários e seus resultados foram incluídos nesta pesquisa. Esta fase corresponde a etapa do *survey* administrando e pontuando o instrumento;

*Passo 7. Analisar e apresentar o relatório de pesquisa.*

A análise dos resultados será feita de duas formas: (i) a análise quantitativa

apoiada por grancos para responder as QP1 e QP2. (ii) E para a QP3 será utilizada a teoria fundamenta em dados [Strauss e Corbin, 1998]. A ênfase na teoria fundamentada em dados é a geração de uma nova teoria. Isto se manifesta de tal maneira que em vez de começar com uma teoria pré-concebida em mente, a teoria evolui durante a pesquisa e é um produto da interação contínua entre coleta e análise desses dados [Goulding, 2002]. De acordo com Strauss e Corbin (1998), a teoria fundamentada em dados é mais parecida com o fenômeno que está realmente acontecendo do que se fosse montada a partir de uma série de conceitos com base na experiência ou através da especulação.

O processo analítico envolve uma estratégia de codificação com os seguintes passos: (1) o processo de quebrar entrevistas, (2) observações e outras formas de dados apropriadas são categorizadas em unidades distintas de significado que são rotuladas para gerar conceitos. Estes conceitos são inicialmente agrupados em categorias descritivas. Os conceitos são então reavaliados para observar as suas inter-relações e, através de uma série de passos analíticos, são gradualmente sumarizados na ordem mais elevada, ou uma categoria central subjacente, o que sugere uma teoria emergente.

Neste caso, a teoria fundamentada foi escolhida como o método de investigação, pelas seguintes razões:

- Devido a falta de uma teoria de como ocorre a melhoria de processo de software ágil em empresas brasileiras que procuram aumentar sua competitividade utilizando modelos de qualidade como o CMMI e MPS.BR.
- Para estabelecer diretrizes para a condução indutiva de uma pesquisa focada na geração de teoria.
- Essa teoria é conhecida pela sua aplicação ao comportamento humano e suas percepções sobre o ambiente [Martin e Turner, 1986]. Neste caso, o desenvolvimento de software é um trabalho intensivo e a melhoria de processo de software depende fortemente do fator humano.
- É uma metodologia estabelecida e possui credibilidade em

disciplinas sociológicas e de saúde [Sheldon, 1998] e vem em crescimento na área de TI.

Além disso, este estudo busca entender a dimensão de desenvolvimento de software na prática. Na perspectiva da melhoria de processo de software o papel dos atores individuais, o ambiente, e contexto pesam sobre como o MPS é praticado. Então, facilitar a coleta e análise das experiências humanas e as inter-relações associadas com outros atores humanos, juntamente com fatores situacionais e contextuais, são os pontos fortes desta metodologia.

Para entender melhor a teoria fundamentada, é preciso entender o conceito de amostragem teórica [Straus e Corbin, 1998]. Amostragem teórica se refere ao processo de coleta, codificação e análise de dados, enquanto simultaneamente é gerada a teoria. Entrevistas, formais e informais, estão no cerne do processo da coleta de dados. Como o pesquisador não sabe com antecedência para onde a teoria vai para conduzi-lo, apenas uma amostra inicial pode ser planejada.

Com base na teoria emergente, os investigadores podem alterar a lista de perguntas para refletir mais de perto sobre as novas categorias emergentes. Com base nestas categorias os pesquisadores podem optar por entrevistar certos tipos de pessoas físicas ou procurar outras fontes de dados. Como os conceitos de categorias continuam a surgir, a amostragem teórica torna-se um processo em constante mudança [Straus e Corbin, 1998].

Assim pesquisadores se engajam em uma "comparação constante" entre os dados analisados e a teoria emergente. Esse processo continua até que exista a saturação teórica, ou seja, os dados adicionais que estão sendo coletados parem de fornecer novos conhecimentos sobre as categorias já encontradas.

A análise dos dados foi realizada a partir das transcrições das entrevistas. A partir da transcrição foram atribuídos códigos ao texto. Neste contexto, códigos são as palavras mais utilizadas nas entrevistas que estejam ligadas ao contexto da MPS. Os códigos representam conceitos que mais tarde tornam-se parte da teoria. Um código retirado de uma transcrição é conhecido como um código de *in-vivo*, e estes são especialmente importantes na medida em que vêm

diretamente dos entrevistados, não exigindo interpretação por parte do pesquisador. A partir das entrevistas iniciais, uma lista de códigos emerge e esta lista é então usada para codificar entrevistas subsequentes. No final do processo um grande número de códigos já pode ser identificado [Straus e Corbin, 1998].

Outra parte da codificação é chamada de axial. A codificação é o processo de relacionar categorias para suas subcategorias, e é denominada axial porque a codificação ocorre em torno do eixo de uma categoria que é ligada a subcategorias [Straus e Corbin, 1998].

Por fim deve ser realizada a codificação seletiva que é o processo de integrar e refinar a teoria. Como as categorias são apenas descrições dos dados que devem ser desenvolvidas para formar a teoria, o primeiro passo é o de identificar a central, ou núcleo da categoria em torno do qual a teoria será construída. À medida que a categoria núcleo funciona como o eixo para todas as outras categorias identificadas, esta deve ter um papel central em que todas as outras categorias devem referir-se a ela [Straus e Corbin, 1998].

Uma atividade comum na pesquisa ação é conhecida como *Memoingö*, que é o processo contínuo de fazer anotações e ideias de questões que ocorrem ao analista durante o processo de coleta de dados e análise [Schreiber, 2011]. Normalmente, as ideias são registradas durante o processo de codificação e os memorandos ajudam a armazenar a teoria que emerge e são escritos constantemente durante o processo da teoria fundamentada. Memorandos podem assumir a forma de declarações, hipóteses ou perguntas. Na última parte do estudo, na sequência de codificação e análises extensivas, os memorandos se tornam cada vez mais teóricos e agem como blocos de construção para o relatório final. Nesta pesquisa, o questionário do apêndice D avalia o processo de evolução. Inicialmente apenas uma pergunta aberta estava presente (Questão 13) e esse questionário foi evoluindo até a sua versão final com quatro questões abertas.

### 3.3. SUMÁRIO DO CAPÍTULO

O principal método de pesquisa adotado foi o estudo de caso que foi complementado por outros dois métodos: (i) *survey* e (ii) teoria fundamentada em dados. O estudo de caso foi baseado na estrutura proposta Brereton et al. (2008) que é composto por sete fases. Uma parte dos dados coletados foram realizados com a utilização do método de pesquisa *survey*, seguindo as referências de Kitchenham e Pfleeger (2002a até 2002d) e o ciclo de vida do *survey* foi realizado de acordo com os passos equivalentes no estudo de caso.

As coletas de dados realizadas a partir do *survey* foram realizadas em três momentos: (i) antes do início do programa de MPS; (ii) imediatamente após as avaliações do modelos de qualidade; e (iii) após seis meses das avaliações dos modelos de qualidade, sendo estas últimas repetidas a cada seis meses. Todos os questionários continham questões fechadas que auxiliaram a responder as questões de pesquisa QP1 e QP2. Os questionários que foram enviados por último traziam questões abertas que responderiam a questão QP3.

As questões fechadas foram analisadas de forma quantitativa apoiada pela criação de gráficos que foram comparados com os resultados do MPS. O objetivo era de levantar indícios sobre as vantagens e desvantagens da utilização do MPS baseado em métodos ágeis, se comparado com as empresas brasileiras que apenas se baseiam no MPS.BR para realizar seu programa de melhoria de processo. As questões abertas foram analisadas a partir da técnica *teoria fundamentada em dados*, que se apresenta como uma abordagem qualitativa para análise dos dados advindos de entrevistas.

## 4. Análise dos Resultados

Neste capítulo serão apresentados os resultados da pesquisa. Mostraremos aqui os resultados de acordo com as questões de pesquisa previamente definidas.

### 4.1. QP1 - É POSSÍVEL EXECUTAR PROGRAMAS DE MELHORIA DE PROCESSO DE SOFTWARE BASEADOS EM METODOLOGIAS ÁGEIS COM O OBJETIVO DE CERTIFICAÇÃO CMMI NÍVEL DOIS E MPS.BR NÍVEL F?

Oito empresas responderam ao questionário que respondia a questão de pesquisa QP1. Destas oito, seis foram avaliadas com sucesso e duas não conseguiram terminar o programa de MPS com a certificação. Assim, os resultados do universo de empresas pesquisadas apresentam uma taxa de sucesso de 75% nos programas de MPS.

Não há muitas pesquisas recentes sobre a taxa de sucesso de iniciativas de Melhoria de Processo de Software, Herbsleb et al. (1997) apontam que 2/3 das iniciativas de MPS para fins de obtenção de certificações CMM falham. Krasner (1999) aponta que metade das tentativas de melhorias de processos em geral na organização tende ao fracasso enquanto Gupta (2004) afirma que 60% das iniciativas de melhoria de processo baseadas no *six sigma* [Pande e Holpe, 2001] também tendem a fracassar.

A Figura 10 apresenta os resultados comparativos entre as abordagens; os resultados desta tese estão identificados pelo rótulo *Santana*. Os resultados apresentados na Figura 10 não podem ser generalizados devido ao pequeno número de empresas pesquisadas (oito), entretanto, existe a indicação de que este tipo de implementação de MPS pode apresentar resultados significativamente melhores do que outras iniciativas pesquisadas.

Algumas opiniões relevantes surgiram dos questionários entregues tanto pelas equipes que foram bem sucedidas quanto pelas equipes que falharam na obtenção das certificações de qualidade sobre especificamente esta questão de pesquisa.

### Iniciativas de MPS Bem Sucedidas em %

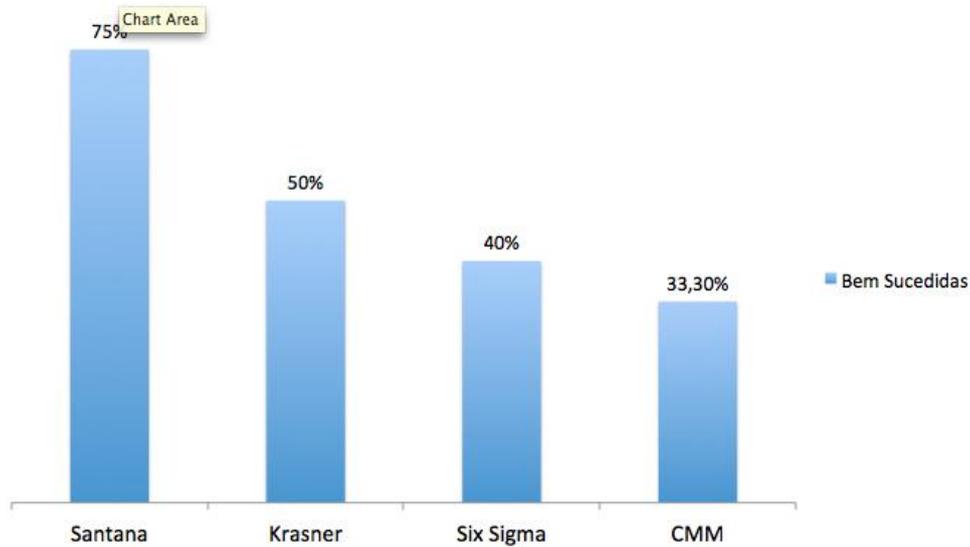


Figura 10 - Taxa de Sucesso em Programas de MPS.

Questionários enviados por duas das empresas continham relatos que é percebida uma clara diferença entre a MPS ocorrida na equipe de desenvolvimento e no MPS ocorrido no núcleo de negócios da organização. Os relatos estão transcritos a seguir.

*“Fica difícil conciliar o pensamento ágil que existe dentro do time com o excesso de documentação e atividades exigidas pela alta gestão. São muitos planos, reuniões, auditorias e relatórios que interrompem as atividades da equipe.”* Líder de Equipe da Empresa .

*“O time trabalha de acordo com o Scrum. Todo o nosso modo de trabalho foi alterado, mas a alta gestão procura controle e repetição. Essa diferença de cultura faz com que parte do trabalho do time seja para atender a gestão, e não ao cliente.”* Gerente de Qualidade da Empresa .

Esses relatos trazem a tona que mesmo iniciando o MPS com a utilização de abordagens ágeis, estas organizações ainda restringem a sua iniciativa de agilizar o programa de MPS para o nível operacional, enquanto o nível de negócio pouco se altera. Essa abordagem não muito difere da abordagem *“O quê versus como”* apresentada nos mapeamentos existentes.

Existe a possibilidade deste fato estar ocorrendo devido ao nível de maturidade desejado pelas empresas, Nível 2 do CMMI e Nível F do MPS.BR.

Estes níveis estão preocupados com os projetos e isto é controlado na organização dos times e não na alta-gestão.

#### 4.2. QP2 ó QUAIS OS BENEFÍCIOS OBSERVADOS POR ESTAS ORGANIZAÇÕES AO SE ADOTAR A MELHORIA DE PROCESSO BASEADA EM MÉTODOS ÁGEIS? E EM LONGO PRAZO?

Os resultados desta questão de pesquisa serão divididos em quatro grupos para melhor explicação dos resultados.

##### **Grupo 1 ó Indicadores Gerais**

Foram considerados com indicadores gerais os resultados do tempo de implementação, retorno de investimento e gasto com a implementação do MPS relativo ao faturamento total da companhia. Os resultados são mostrados na Tabela 12.

**Tabela 12 – Comparativo dos Indicadores Gerais das Empresas pesquisadas x iMPS**

<b>Indicador</b>	<b>Resultados iMPS</b>	<b>Resultados das Empresas Pesquisadas</b>
<i>Tempo de Implementação</i>	Entre 12 e 16,5 Meses (MPS.BR)	13 Meses (MPS.PR) e 15 Meses CMMI.
<i>ROI maior que 1</i>	Entre 50% e 60%	66%
<i>Custo da Implementação x Faturamento Bruto</i>	Entre 2,8% e 10% do Faturamento Bruto	4% do Faturamento Bruto

Os resultados apresentados na Tabela 12 sugerem que a implementação do MPS baseado em métodos ágeis, realizado nas empresas estudadas nesta pesquisa, está dentro da faixa normal da implementação do MPS.BR, mostrada pelo iMPS, quando observado o tempo e o custo da implementação. Quanto ao ROI, o resultado das empresas pesquisadas se mostram ligeiramente melhores do

que as empresas que responderam o iMPS. Vale ressaltar, novamente, que estes números não apresentam uma generalização, pois, os 66% obtidos neste estudo são dados de apenas duas empresas.

Mas, observando os resultados dos questionários abertos, existe um relato que pode ser destacado por apresentar uma peculiaridade:

*“A equipe demorou a perceber os ganhos trazidos pela iniciativa do MPS.BR dentro da organização, tanto que estendemos um pouco o tempo de implementação já que esta representava um custo mais acessível se comparado com o custo de uma possível reavaliação em caso de reprovação...”* Gerente de Qualidade da Empresa .

Este relato é curioso por que no iMPS é dito que o custo da implementação é, em média, quatro vezes maior do que o da avaliação em si. Isso significa que na implementação do MPS.BR baseado em métodos ágeis, há indícios de que uma parte considerável do investimento realizado no programa é utilizada no momento da avaliação. Vale ressaltar que as empresas pesquisadas passaram também por uma avaliação CMMI que custa entre quatro a seis vezes mais do que uma avaliação MPS.BR, e talvez seja por isso que o gasto deixa de ser concentrado no início do programa para ser concentrado nas duas avaliações (MPS.BR e CMMI). Contudo, considerando que o valor da avaliação CMMI também entra no valor total da implementação (4% do faturamento bruto), há aqui o indício que o custo desta de implementação de MPS baseada em métodos ágeis pode ser comparado aos níveis mais baratos relatados no iMPS (2,8% no ano de 2009).

## **Grupo 2 ó Relação dos Indicadores de Desempenho entre antes e depois das avaliações.**

Estes indicadores representam a evolução das organizações entre os períodos anteriores ao início do processo de MPS e imediatamente após a avaliação. Estes indicadores serão comparados com os resultados do iMPS como base comparativa. As Figuras 11 a 15 apresentam a compilação entre as

empresas pesquisadas e o iMPS.

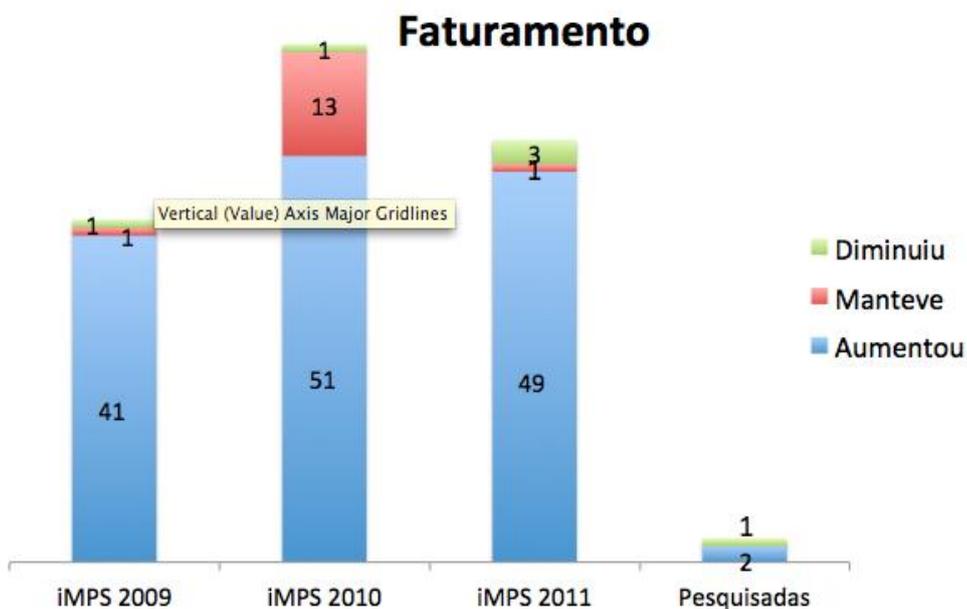


Figura 11 – Comparativo do faturamento das empresas iMPS e as empresas pesquisadas.

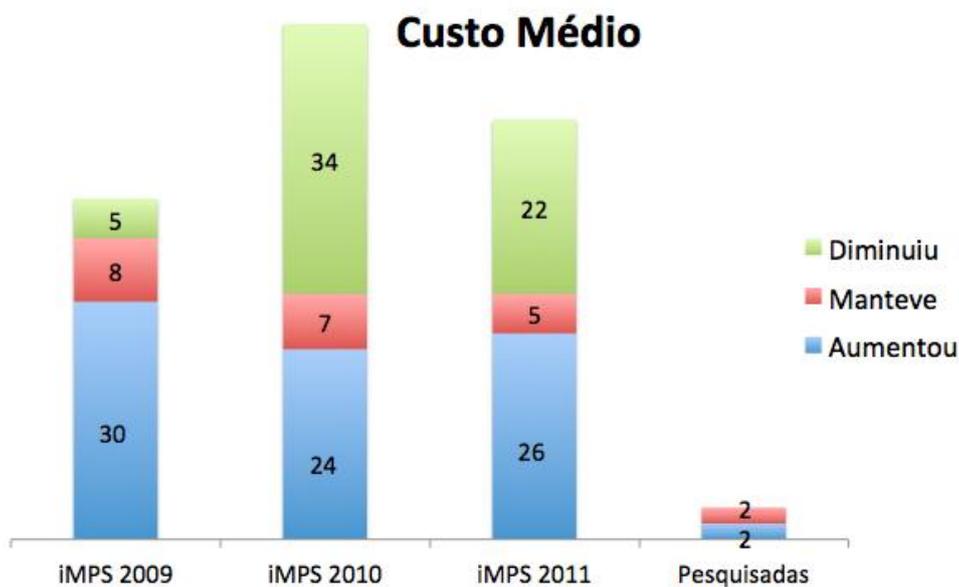


Figura 12 – Comparativo do custo médio por projetos das empresas iMPS e as empresas pesquisadas.

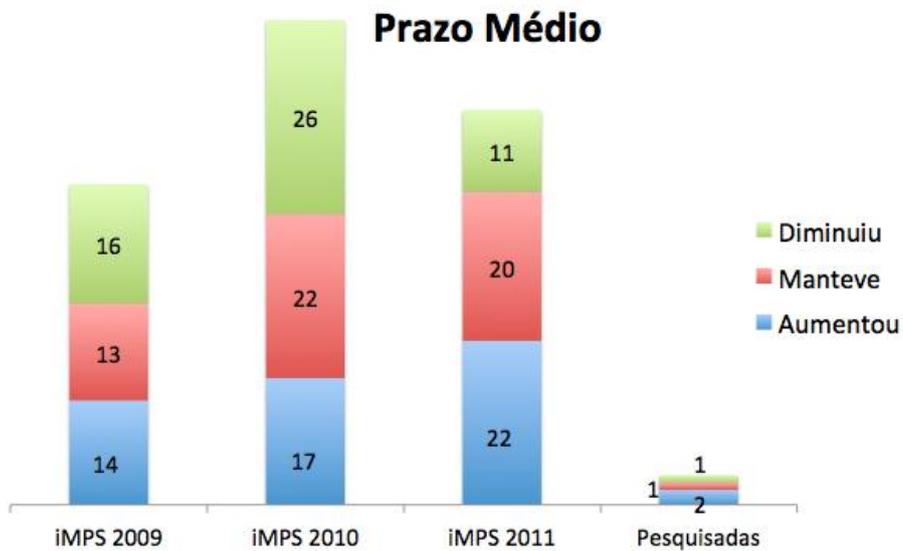


Figura 13 – Comparativo do prazo médio por projetos das empresas iMPS e as empresas pesquisadas.

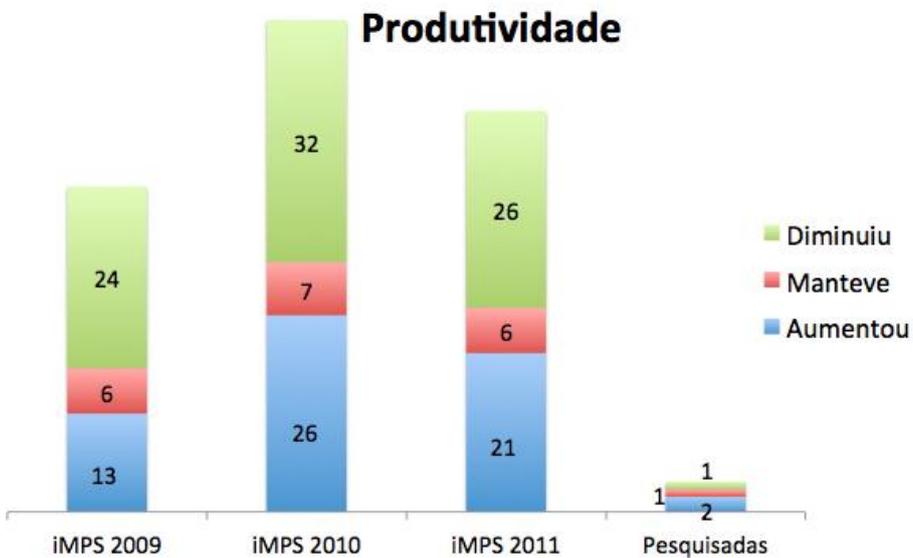


Figura 14 – Comparativo da produtividade das empresas iMPS e as empresas pesquisadas.

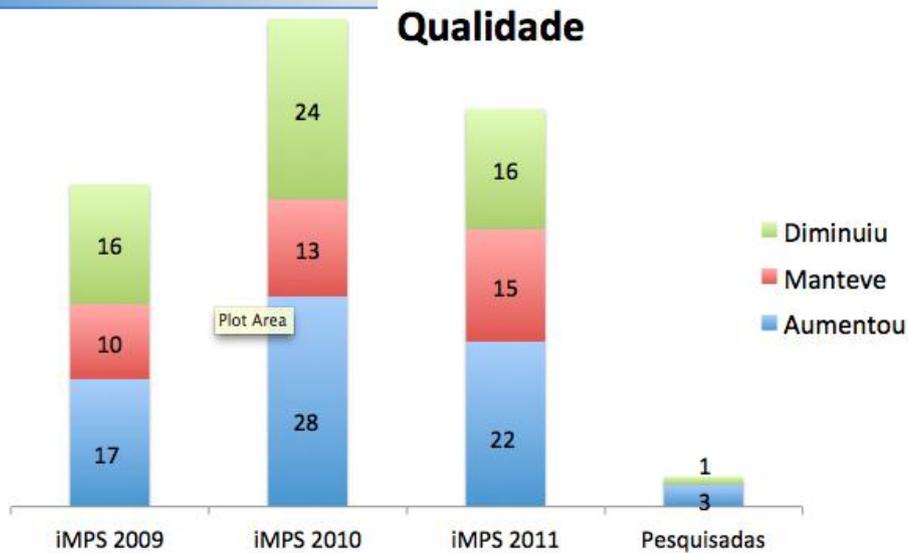


Figura 15 – Comparativo da qualidade das empresas iMPS e as empresas pesquisadas.

Para auxiliar na análise, também foram criados os dados percentuais de cada indicador. Estes são apresentados nas Figuras de 16 a 20. Ressaltando que estes dados percentuais são aqui apresentados apenas como complemento não podendo ser considerados conclusivos devido ao tamanho pequeno da amostra pesquisada.

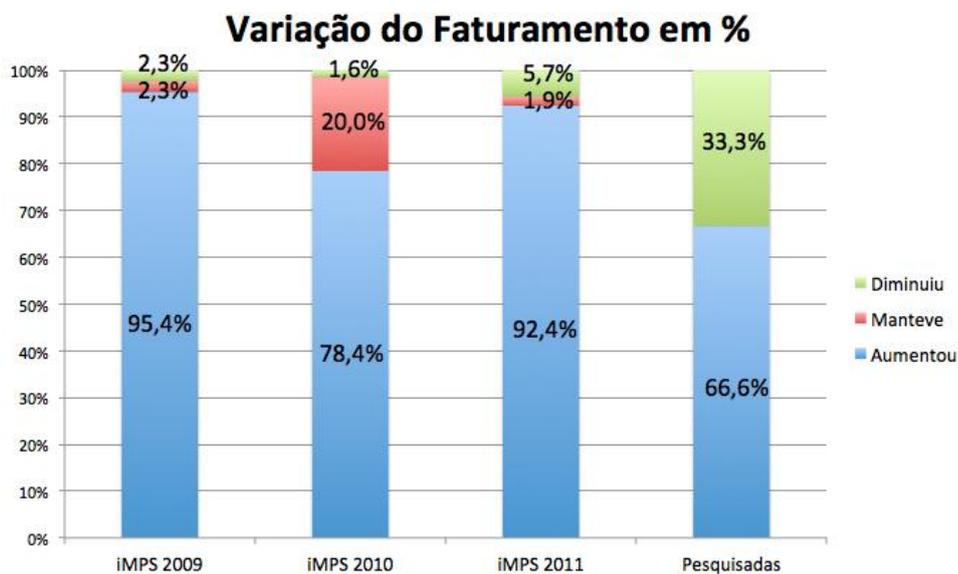


Figura 16 – Comparativo do faturamento das empresas iMPS e as empresas pesquisadas em porcentagem.

### Varição do Custo Médio em %

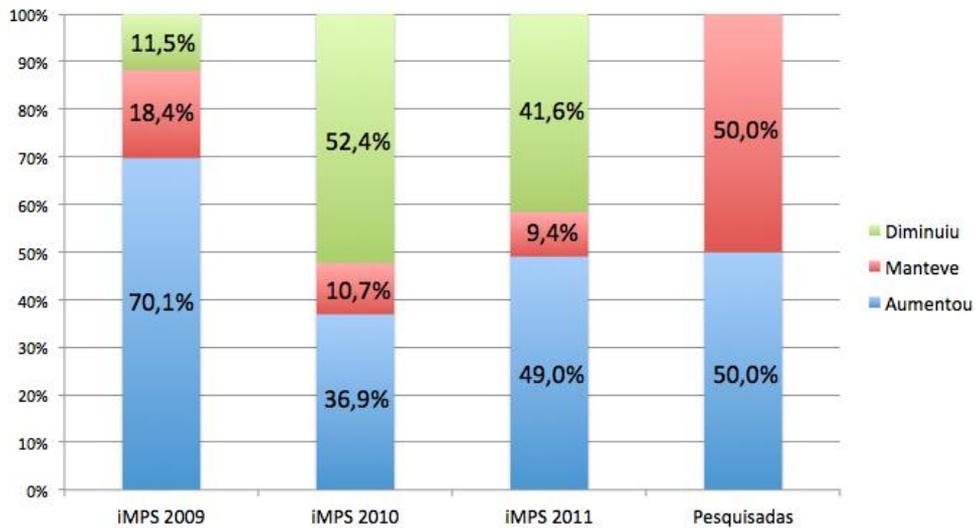


Figura 17 – Comparativo do custo médio das empresas iMPS e as empresas pesquisadas em porcentagem.

### Varição do Prazo Médio em %

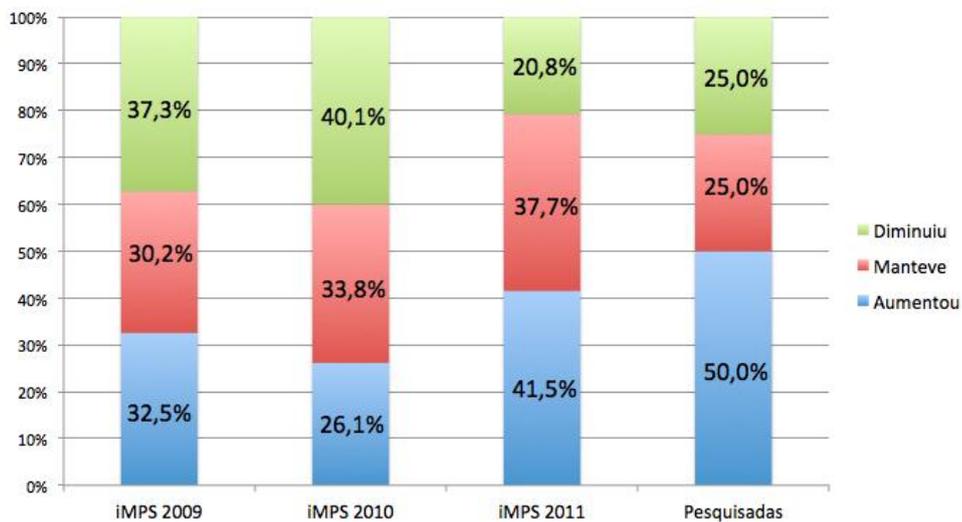


Figura 18 – Comparativo do prazo médio das empresas iMPS e as empresas pesquisadas em porcentagem.

### Variação da Produtividade em %

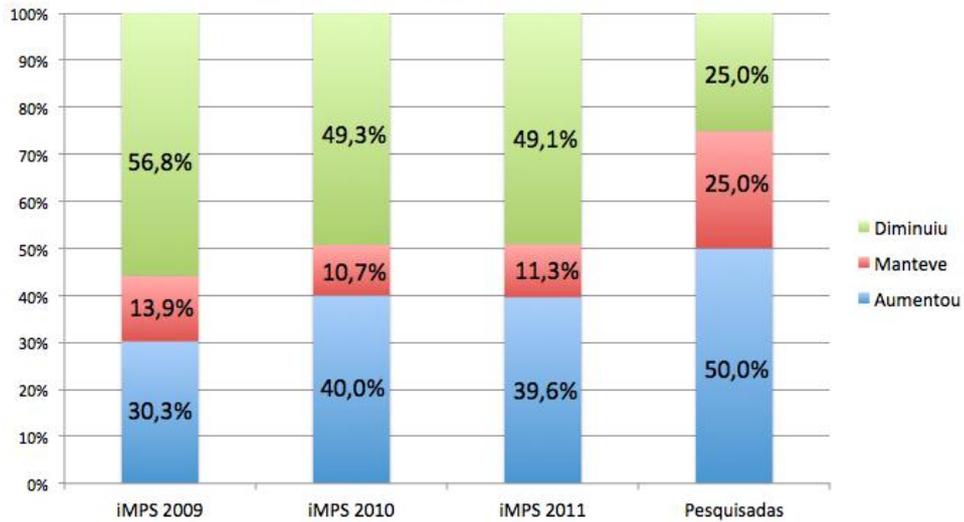


Figura 19 – Comparativo da produtividade das empresas iMPS e as empresas pesquisadas em porcentagem.

### Variação da Qualidade em %

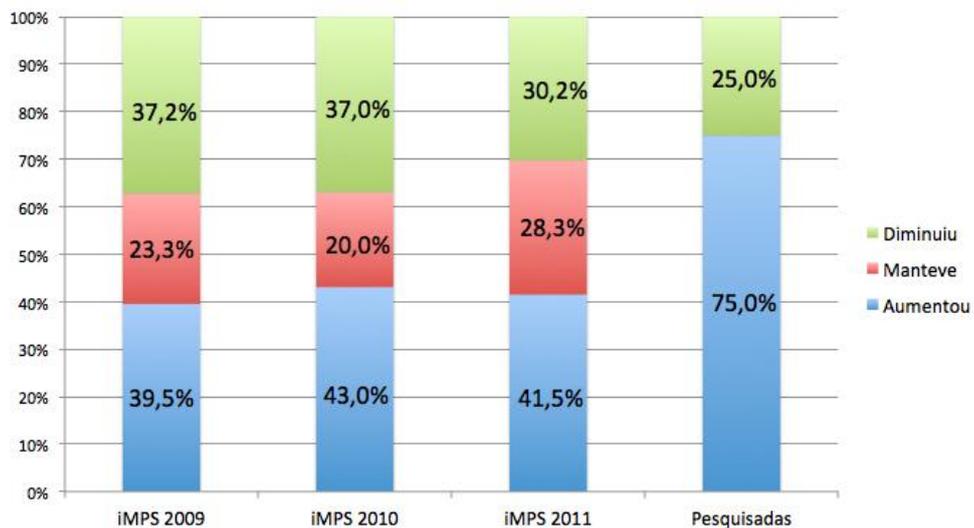


Figura 20 – Comparativo da qualidade das empresas iMPS e as empresas pesquisadas em porcentagem.

O grupo pesquisado apresentou nos seus indicadores percentuais

resultados mais interessantes. Por exemplo, o aumento da produtividade, que ocorre em no máximo 40% das empresas investigadas pelo iMPS, enquanto no grupo de empresas pesquisado esse percentual é de 50%. O grupo de empresas pesquisadas também apresentava o menor percentual onde ocorreu a diminuição (25%). O aumento da qualidade é outro indicador onde os resultados das empresas pesquisadas se apresentam melhores do que as empresas pesquisadas iMPS (75% versus 43%)

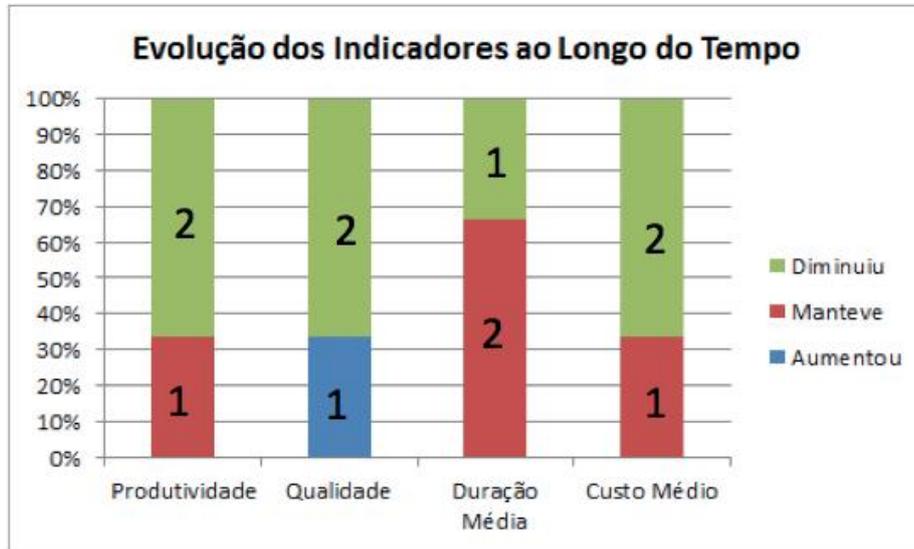
Outros indicadores apresentaram resultados piores nas empresas pesquisadas. Por exemplo, no grupo de empresas pesquisadas o custo médio por projeto aumentou em 50% delas, este número está em consonância com as empresas do iMPS cujo o aumento do custo médio ocorre entre 70,1% a 36,9% delas. Entretanto, nenhuma das empresas pesquisadas diminuiu o custo médio por projeto, enquanto nas empresas iMPS a diminuição do custo se deu em 11,5% a 52,4%.

Esse fenômeno fica ainda mais claro quando observamos o prazo médio dos projetos, pois o número de empresas pesquisadas que aumentaram seus prazos foi de 50%. Este número é superior ao indicador observado no iMPS entre 2009 e 2011. Por fim, o número de empresas pesquisadas onde o faturamento aumentou (66%) é menor do que o número de empresas que aumentaram o seu faturamento de acordo com o iMPS (entre 78,4% e 95,4%).

### **Grupo 3 ó Evolução dos Indicadores de Desempenho das organizações em longo prazo.**

Estes indicadores representam a evolução das organizações após as certificações CMMI e MPS.BR. Neste momento, não será usada outra base de dados como grupo de comparação, e sim será avaliada a evolução das empresas, observando seus próprios indicadores de forma periódica. A cada seis meses foram coletados os indicadores de produtividade, custo e tempo médio do projeto, faturamento e qualidade do produto. Esta coleta consistia em verificar se houve aumento ou diminuição do mesmo. A evolução destas empresas é

mostrada na Figura 21.



**Figura 21 – Evolução dos Indicadores das Organizações após as avaliações dos modelos de Maturidade.**

Percebemos que em média as empresas vão ficando menos produtivas, com uma qualidade pior, com projetos mais caros e mais longos. Isso pode ser justificado pela erosão do processo. Isto leva a organização a abandonar o processo criado para a organização durante a implementação do MPS em direção a um processo mínimo e *Ad-hoc* sobre o funcionamento da organização. Detalhes sobre a erosão de processo estarão presentes na próxima subseção a seguir.

#### **Grupo 4 ó Erosão de Processo.**

Para avaliar a erosão do processo foi utilizada a questão 16 nos questionários do 3º Levantamento de Informações (Apêndice D). Vamos dividir as explicações em duas partes, que são sobre a erosão de processo aderente aos modelos de maturidade e a erosão do processo ágil, todos baseados em Scrum, das organizações pesquisadas.

Além de todo o contexto da erosão de processo explicado na Seção 2.5 desta tese, é preciso apresentar como a erosão de processo se apresenta nesse conceito de melhoria de processo de software. Segundo Coleman e Connor,

(2008) toda iniciativa de MPS passa por três fases distintas que é a Formação do Processo, a Evolução do Processo e a Erosão do Processo.

A formação do processo é a etapa onde, devido aos objetivos de negócio da organização, um novo processo baseado nesses novos objetivos de negócios é proposto. A evolução deste é o momento onde a organização leva-o do estado atual até ao estado desejado pela organização. Após a estabilização deste novo processo, começa o processo de erosão onde algumas práticas são abandonadas e este tende a recuar a um processo mínimo com o básico para o funcionamento da organização [Coleman e Connor, 2008].

As empresas pesquisadas mostraram um padrão comum para a erosão do processo aderente ao CMMI e ao MPS.BR. Primeiro algumas práticas de gestão de projetos foram abandonadas, a seguir, é abandonada a garantia da qualidade. As Figuras a seguir representam erosões do processo no CMMI e MPS.BR reportadas por algumas empresas.

As Figuras 22 e 23 apresentam as empresas que mais e menos sofreram erosão no processo CMMI. Para o CMMI foi verificada a erosão nas áreas de processo: Planejamento de Projetos (PP), Monitoramento e Controle de Projetos (PMC), Gerenciamento de Requisitos (REQM), Medição e Análise (MA), Gerência de Configuração (CM) e Garantia da Qualidade (PPQA). A empresa que sofreu a maior erosão (Figura 22) do processo abandonou cerca de 30% das práticas do CMMI. A que sofreu a menor erosão (Figura 23) abandonou 6%.

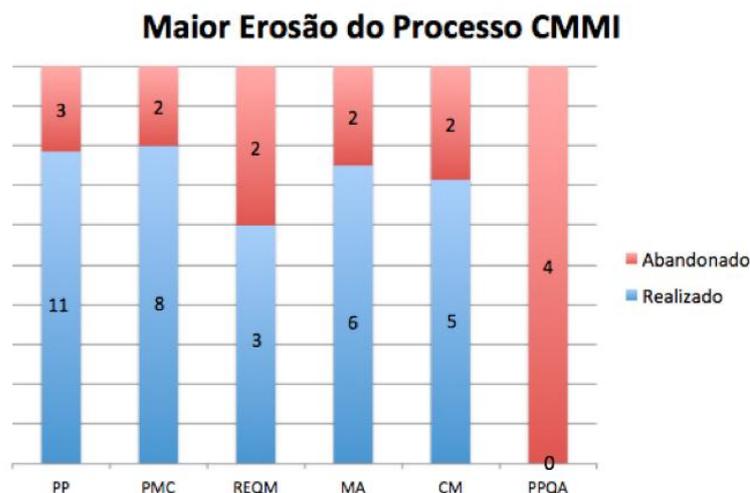


Figura 22 – Resultados da empresa que sofreu a maior erosão de processo para o CMMI

### Menor Erosão do Processo CMMI

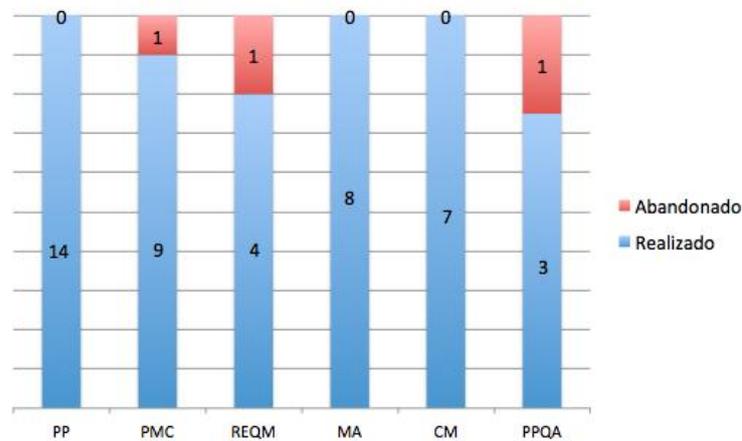


Figura 23 – Resultados da empresa que sofreu a menor erosão de processo para o CMMI

Já a maior e menor erosão verificadas no processo MPS.BR são apresentadas nas Figuras 24 e 25, respectivamente. Para o MPS foram verificadas as áreas de processo: Gestão de Projetos (GPO), Gestão de Requisitos (GRE), Gerência de Configuração (GCO), Medição (MED) e Garantia da Qualidade (QGA).

### Maior Erosão do Processo MPS.BR

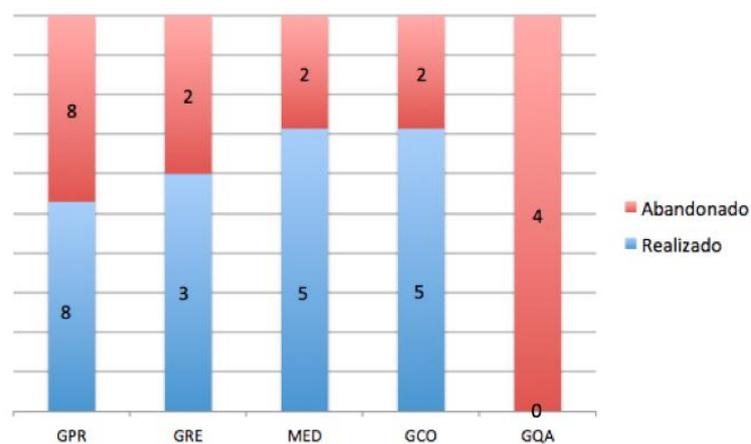
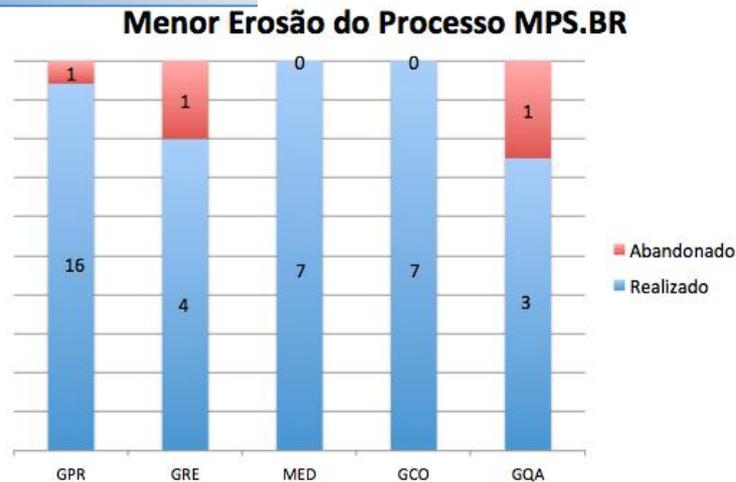
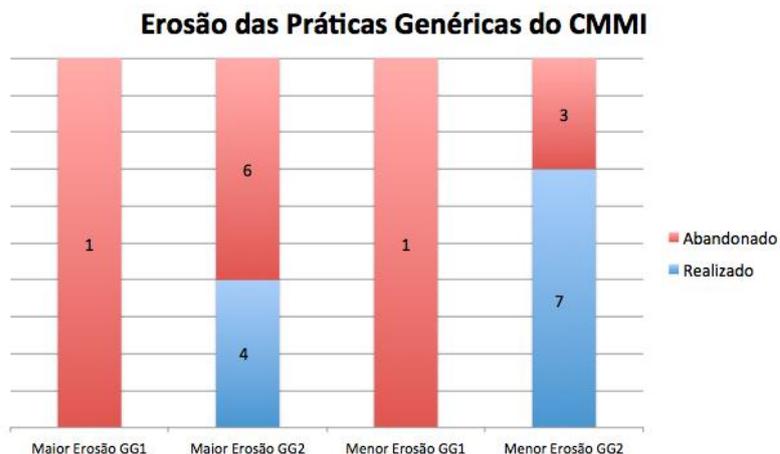


Figura 24 – Resultados da empresa que sofreu a maior erosão de processo para o MPS.BR



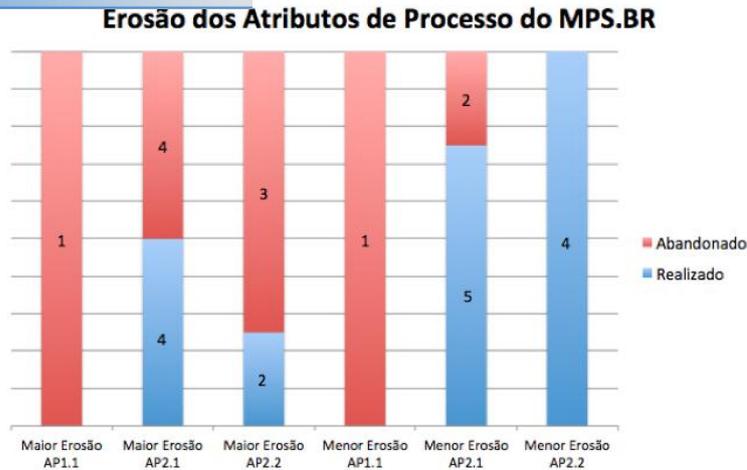
**Figura 25 – Resultados da empresa que sofreu a menor erosão de processo para o MPS.BR**

Se considerarmos os elementos norteadores (Práticas Genéricas do CMMI e Atributos de Processo do MPS.BR) também podemos ver a maior e menor erosão de processo sofrida por eles nas Figuras 26 e 27, respectivamente.



**Figura 26 – Erosão das Práticas Genéricas do CMMI**

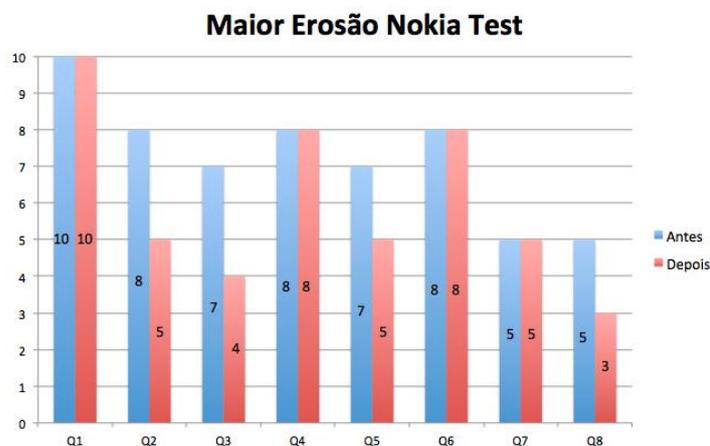
Podemos perceber que a erosão nos elementos norteadores é significativamente maior do que nas práticas específicas. Como estes elementos refletem muito da cultura da organização, uma das possíveis causas da maior erosão apresentada nesta parte é que falta alinhamento estratégico da equipe de desenvolvimento para com o negócio da organização. Possivelmente, por esta razão, estas práticas sofrem erosão de forma mais acentuada do que as práticas que refletem as atividades técnicas.



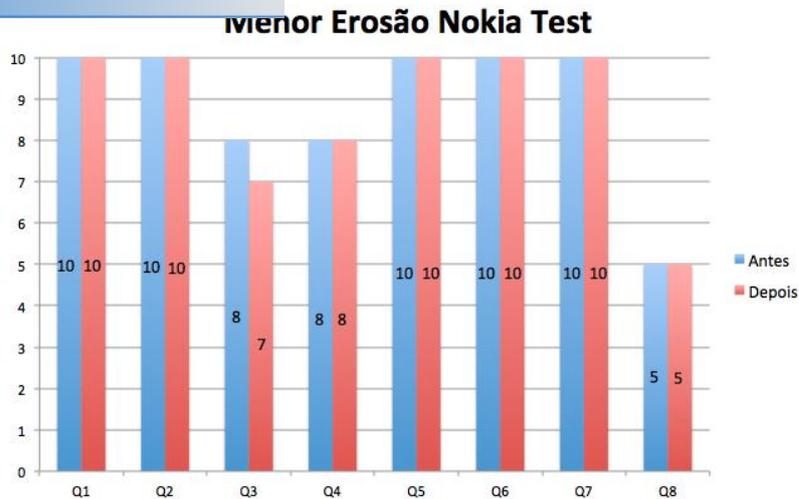
**Figura 27 – Erosão dos Atributos de Processo do MPS.BR**

A erosão do processo referente ao Scrum foi identificada a partir do preenchimento de questionários do *Nokia Test* [Anexo A] proposto por Sutherland (2009). O *Nokia Test* tem por objetivo avaliar como as equipes que adotam Scrum estão executando o processo. O questionário é composto por oito perguntas, e para cada uma é atribuída uma nota de 0 a 10. A partir das notas dadas pelos membros da própria equipe uma média é atribuída, e essa média final é a pontuação da equipe para o *Nokia Test*.

Nesta pesquisa, a avaliação *Nokia Test* foi conduzida com o objetivo de avaliar a erosão dos processos Scrum, já que ele mede de forma quantitativa como os times estão o adotando. As Figuras 28 e 29 apresentam a maior e a menor erosão sofrida em processos Scrum, respectivamente.



**Figura 28 – Maior Erosão de Processo apontada pelo Nokia Test.**



**Figura 29 – Menor Erosão de Processo apontada pelo Nokia Test.**

O resultado é que a Erosão do Processo Scrum é ainda menor do que a Erosão percebida nas empresas CMMI e MPS.BR. Este fator reforça a ideia de que os processos técnicos sofrem uma erosão menor do que aqueles de negócio.

#### 4.3. QP3 ó COMO A MELHORIA DE PROCESSO ÁGIL òINDUTIVAò ESTÁ RELACIONADA À MELHORIA DE PROCESSO òPRESCRITIVAò ADOTADA PELO CMMI E MPS.BR?

Para auxiliar a codificação exigida pela teoria fundamentada em dados, foi montada inicialmente uma nuvem de palavras a partir de todos os questionários recebidos para identificar os códigos mais usados. A Figura 30 mostra a nuvem de palavras resultante dos questionários.

Nela podemos ver que os códigos: negócios, cultura, alinhamento organizacional, contexto, aprendizado e erosão, foram os mais citados e a partir desse mapa podemos considerar o código ònegóciosò como central para a avaliação da teoria fundamentada.

É preciso levar em consideração que o código mudança foi retirado desta representação de nuvem de palavras por que a palavra aparecia em muitos contextos diferentes, tais como, mudança organizacional, mudanças de requisitos, mudanças de procedimentos, mudanças de comportamento, mudanças

de visao, mudança de ferramentas e sua aparição nesta representação seria desproporcional quando vários contextos estariam representados pelo mesmo código. Se o código mudança fosse considerado, teríamos a nuvem de palavras apresentada na Figura 31.

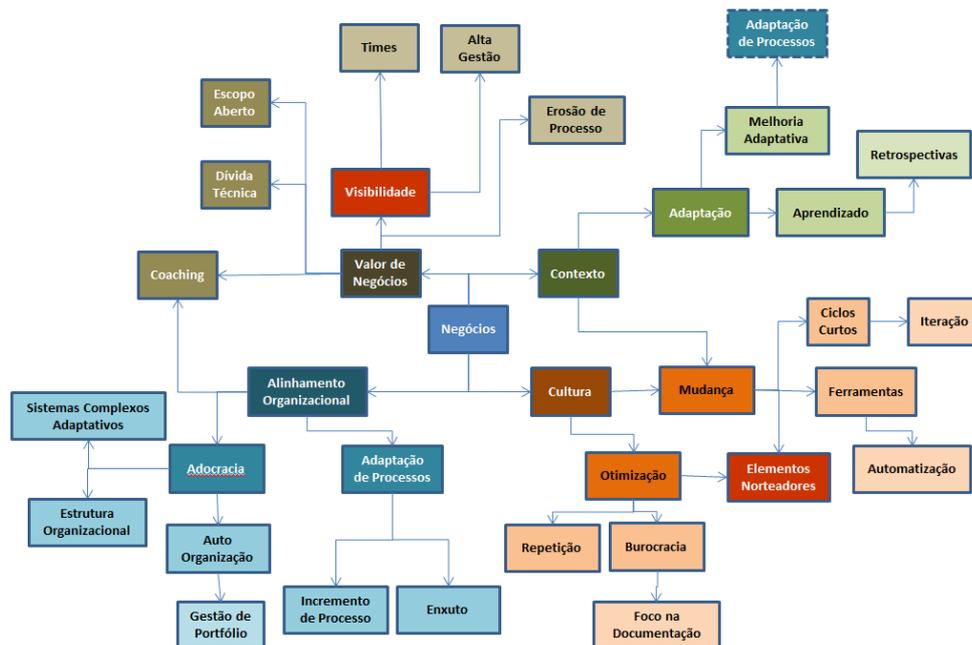


Figura 30 – Nuvem de palavras dos questionários recebidos pelas empresas sem considerar o código mudança.



**Figura 31 – Nuvem de palavras dos questionários recebidos pelas empresas considerando o código mudança.**

A partir da identificação dos códigos centrais encontrados na nuvem de palavras, foi realizada uma nova análise dos questionários para identificar o relacionamento destes códigos. Desta forma, foram identificados alguns relacionamentos específicos apresentados na codificação coaxial da teoria fundamentada. Esta representação é apresentada na Figura 32.



**Figura 32 – Codificação coaxial dos códigos apresentados nos questionários**

Como citado anteriormente, o código negócios se mostrou o central na avaliação dos questionários. Outros quatro códigos se mostraram também relevantes, e são eles: (i) cultura, (ii) contexto, (iii) alinhamento organizacional e (iv) valor de negócios.

Então, será apresentado aqui como o MPS indutivo aparece nas organizações CMMI. Essa apresentação será dividida por grupo de códigos. Além do código central (Negócios), outros quatro códigos detalhados serão: Contexto, Cultura, Valor de Negócios e Alinhamento Organizacional. Então, esta codificação coaxial será detalhada a partir destes cinco códigos, o código central (negócios) e os quatro destacados.

### **Código Central: Negócios.**

Muitos dos entrevistados colocaram que a questão central do MPSágil é que ele não acontece no ponto de vista dos negócios. Normalmente as empresas iniciam seu processo de MPS utilizando a abordagem ágil Scrum, mas essa mentalidade se restringe apenas ao time e não se estende aos níveis mais altos de

gestão, como e passado no relato a seguir:

*“O Scrum começa a funcionar muito bem quando aplicado à equipe de desenvolvimento. Logo o time aprende a responsabilidade do autogerenciamento e a liberdade para inovar que temos nas retrospectivas. Entretanto, o time não pode interferir em nada do que é externo a ele e isso impede que este até melhore sua própria estrutura de trabalho, já que a alta gestão possui alguns pontos de controle sob o nosso time que são o planejamento e monitoração do projeto e as auditorias de qualidade e estas são imutáveis. Na prática somos uma caixa no organograma que tem liberdade para executar o seu trabalho, mas sofremos muitas interrupções dos gerentes e SQA’s. Scrum Master da empresa .*

Considerando este e outros relatos, é notado que, ao se iniciar o programa de MPS, as empresas começam adotando o Scrum para a equipe de desenvolvimento, mas esta mentalidade ágil é restringida ao relacionamento entre o time e o cliente para o desenvolvimento do produto específico.

A intenção das empresas é que o desenvolvimento do produto seja acelerado e as mudanças de requisitos tragam menos impacto nos projetos. Entretanto, essas empresas não desejam modificar o seu modelo de negócios para que este também esteja apto a lidar com mudanças e este conflito impacta toda a iniciativa de MPS praticada na empresa, como veremos nos comentários dos outros códigos encontrados.

### **Códigos relacionados ao código “Cultura”**

Podemos perceber que a cultura está ligada a dois códigos (filhos), que são “mudança” e “otimização”. Este antagonismo não é novidade já que Highsmith (1999) há mais de 10 anos atrás apresentava esta diferença. Mas o que causa surpresa é que estas duas formas co-existem em seus extremos dentro da mesma organização, e não que há uma mistura de conceitos em direção ao modelo híbrido.

Na Seção 2.6 foram mostrados alguns mapeamentos e casos de vida real

misturando métodos ágeis e abordagens tradicionais no mesmo processo. A partir dessa mistura se criava o conceito de abordagens híbridas, que não eram tão rigorosas quanto às abordagens tradicionais, nem tão aptas a mudanças quanto os métodos ágeis. Este modelo híbrido se mostra uma solução difícil de ser aplicada. Hignsmith (2002b) afirma que atingir a estabilidade é fácil ó apenas siga os passos, atingir o caos também é fácil ó apenas faça, mas balancear é difícil porque requer um enorme talento gerencial e de liderança.

Pela forma como essa hibridização é proposta, podemos identificar claramente uma divisão na organização entre a mentalidade do time (ágil) e da gestão (tradicional). Em algum momento, estas visões entram em choque, pois a cultura da mudança é baseada em ciclos curtos de respostas, enquanto a cultura otimização é baseada na estabilidade do processo, que é repetido para todos os projetos da organização, no controle baseado na burocracia e na documentação excessiva.

Este conflito destas visões, neste caso, se deu quando se começou a avaliar os elementos norteadores dos modelos de qualidade (práticas genéricas do CMMI e Atributos do Processo do MPS.BR). Isto é relatado por um dos entrevistados.

*õHouve complicação ao fazer a transição daquilo que o time fazia para a aquilo que deveria ser evidenciado, e principalmente as atividades que envolviam a alta gestão e membros externos ao time. Estabelecer políticas organizacionais, planejar os processos de garantia da qualidade e gestão da configuração, controlar os produtos de trabalho que não eram produzidos pelo nosso time, entre outras práticas genéricas, porque a forma de trabalho era completamente distinta. Então tínhamos que parar de fazer nosso trabalho e realizar tarefas alheias ao desenvolvimento do produto para gerar evidências de coisas que nem deveríamos fazerö. Líder de Equipe da Empresa .*

É por este motivo que os elementos norteadores se destacam, porque mostram sinais da cultura organizacional que, neste caso, claramente não se propõe a flexibilizar um processo pesado. Mas sim agilizar o seu chão de fábrica (Mudança), e manter o controle do negócio para produzir mais, em menos tempo

e mais barato (Otimização). Esta realidade não apresenta um ómeio termoö e sim que partes distintas das organizações trabalham de forma diferente.

### **Códigos relacionados ao código òContextoö**

O Contexto também está ligado a dois códigos, que são òmudançaö e òadaptaçãoö. A ligação com a mudança se dá devido ao propósito de qualquer método ágil de apoiar toda e qualquer mudança. Este resultado também não é novidade já que no último questionário *State of Agile Survey* [Versionone, 2011] é apontado que 84% das empresas percebem uma maior facilidade na forma de gerenciar mudanças de prioridades. Sendo que, especificamente nesta pesquisa, o contexto de mudança está quase que exclusivo no time e não no nível de negócios ou alta gestão. Os times estão aptos a lidar com as mudanças de prioridades do cliente, mas não foi mencionado o quanto o contexto dos negócios exigiu mudanças.

A segunda ligação do contexto é com a adaptação, por que os times entendem que mudanças são fatores externos que afetam o time de fora (ambiente) para dentro e que ocorrem independentemente da postura deste time, enquanto a adaptação é como este time se ajusta para continuar o mais produtivo possível devido a estas mudanças.

Estes relatos estão em consonância com a pesquisa de Bakersville (2010) que afirma que existe uma diferença sutil entre adaptar e mudar. Adaptação está focada no ambiente externoedepende de um fluxo livre de informação no ambiente para que decisões realistasbaseadas em informações possam ser tomadas alinhadas com os valores organizacionais. Muitas mudanças nas organizações são feitas em resposta à política interna, em vez de necessidades advindas darealidade externa. Organizações adaptativas devem aprender de fora para dentro, em vez de dentro para fora.

A adaptação está ligada a dois códigos: (1) Melhoria Adaptativa e (2) Aprendizagem. A melhoria adaptativa de processo foi apresentada por Dorling (2010) como uma abordagem que auxilia organizações a identificar como seria a

menor maneira de implementar as melhorias de processos identificadas, e como podemos avaliar a capacidade e disponibilidade para iniciar as melhorias.

Esta abordagem procura a melhoria do processo organizacional que se concentra na adaptação da aprendizagem baseada nos sucessos anteriores e os utiliza como base para o desenvolvimento e otimização de estratégias futuras. Uma organização que emprega a melhoria do processo de adaptação irá fazer melhorias incrementais aos processos em resposta ao ambiente de negócios, que está em constante mudança [Dorling, 2010].

Como parte da melhoria, uma organização precisa escolher uma estratégia de mudança que seja apropriada para o modo de funcionamento organizacional. A experiência mostra que diferentes tipos de trabalhos de melhoramento exigem métodos diferentes para melhorias organizacionais [Dorling, 2010].

Este conceito existe dentro dos times que realmente conduzem a melhoria adaptativa de processos a partir das retrospectivas [Kerth, 2001] realizadas ao final de cada iteração e com os ajustes no comportamento do time. Entretanto, se considerarmos a diferença entre mudanças e adaptação, os times não estão se adaptando, e sim mudando de acordo com os resultados das entrevistas, conforme os comentários a seguir:

*As nossas retrospectivas eram empolgantes nas primeiras iterações, mas depois de poucas iterações nos pegamos resolvendo problemas sobre como iríamos fazer um determinado plano, ou apresentar relatórios ou coletar informações que seriam úteis para o CMMI. Praticamente não mudávamos em virtude do cliente e sim da própria organização e principalmente da geração de evidências do CMMI.* Líder de Equipe da Empresa .

*Em um dado momento abrimos mão de entregar alguns documentos e ainda não fomos questionados pela garantia da qualidade. Parece que esses documentos não acrescentam em nada na organização, apenas nos tomam tempo e esforço.* Desenvolvedor da Equipe .

Parece que esses times realizam uma Melhoria de Processo Orientado a Mudanças ao invés de uma melhoria adaptativa de processo. Os times de fato

fazem memórias incrementais no processo ao fim de cada iteração (Incremento de Processo) baseada em sua experiência, mas a avaliação do negócio que deveria ser realizado na alta gestão não é alinhada com o time de forma iterativa.

Este fato também influencia em outro conceito que está ligado a adaptação que é o aprendizado. Esta estrutura organizacional que possui uma divisão clara entre negócio (O quê) e o time (como) promove uma forma tradicional de aprendizado onde a organização aprende baseado na soma do aprendizado dos seus colaboradores a partir de ciclo longo e de uma iteração. Estes são baseados em lições aprendidas e reuniões pós-morte de projetos [Salo, 2007].

Só que neste ambiente de MPS ágil, esse mecanismo de aprendizagem tradicional não é útil, por que a soma do conhecimento produzido pelos times se mostra um grande repositório técnico e de projetos. Muito embora isso pouco influencie o negócio da organização, pois este conhecimento não considera as influências externas e as reais necessidades de adaptação de negócios.

Assim, a organização tende a uma estrutura de processos repetitiva onde passa pelos mesmos problemas e sua base de conhecimento mostra que situações podem ser resolvidas baseadas em experiências anteriores. Mas, e quando há uma mudança no contexto de negócios? Esta estrutura de otimização realmente apoia essas mudanças? Então é necessária a criação de um novo mecanismo de aprendizado nessa nova estrutura de MPS ágil em que a adaptação é mais importante do que a otimização. Atualmente existe a dificuldade de capturar a experiência dos times e colocar a disposição para a organização. Isso tem sido feito a partir dos registros das reuniões de retrospectiva dos times, mas esse conhecimento não é estruturado para facilitar o seu reaproveitamento.

Mas como incluir o time nas mudanças de negócios? Como transformar o time em uma parte ativa da organização ao invés de uma caixa no organograma, onde ela recebe um conjunto de entradas e produz um conjunto de saídas? Uma observação sobre este aprendizado é identificada em uma das respostas dos questionários:

*“Tínhamos muitos problemas com a nossa política de desenvolvimento*

ae proautos, por que unnamos que alinhar muita coisa com muita gente. Com o Governador, com os secretários, com o fornecedor, com os clientes e com a nossa própria empresa. Só começamos a trabalhar sem esses problemas quando começamos a envolver o time nas reuniões de negócio e o próprio time além de aprender sobre a parte técnica, aprendeu sobre o negócio do cliente, o negócio dos fornecedores e principalmente sobre o negócio desta agência. Foi muito mais produtivo quando o time passou a aprender mais do que a gerência, por que ele tinha uma melhor capacidade criativa para transformar as necessidades de negócio em soluções. Product Owner da empresa .

Então para essa nova abordagem de Melhoria Adaptativa de Processo, MPS Ágil e adaptação, pode ser interessante criar uma estrutura em que o time tenha a oportunidade de aprender sobre a parte técnica e a parte dos negócios.

### **Códigos relacionados ao código ãAlinhamento Organizacionalö**

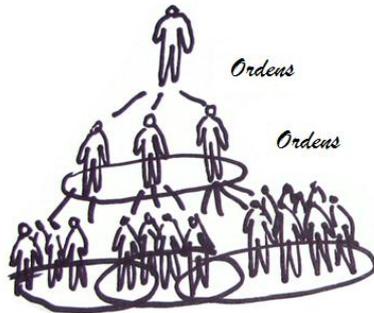
O Alinhamento Organizacional também está ligado a dois códigos, que são: ãadaptação de processosö e ãadocraciaö. Quando tratamos da adaptação de processos, a alta gestão e o time divergem sobre como isso deve ser feito: a alta gestão procura enxugar o processo (*Lean*), enquanto o time realiza a sua melhoria na forma de incremento de processos ao fim de cada iteração.

Ao que concerne a alta gestão, esse enxugamento do processo só é realizado na concepção onde atividades e documentos desnecessários são eliminados. Entretanto, ao longo do desenvolvimento do processo, ela exige que um novo conjunto de documentos sejam produzidos e que novos passos sejam cumpridos para garantir a aderência ao modelo de maturidade. O time inicialmente tira proveito das retrospectivas para encontrar uma melhor forma de trabalhar com o cliente e uma vez chegado neste estágio, os novos incrementos no processo são sugeridos pela alta-gestão, para resolver problemas internos de aderência aos modelos de maturidade. Em suma, a partir de um determinado momento, o time passa a responder às solicitações da alta gestão independentemente de esta ser a melhor solução para ele.

O outro código é a adocracia. Segundo Waterman (1990) a adocracia é um sistema temporário variável e adaptativo, organizado em torno de problemas a ser resolvidos por grupos de pessoas com habilidades e profissões diversas e complementares, constituindo-se como opção à tradicional Departamentalização.

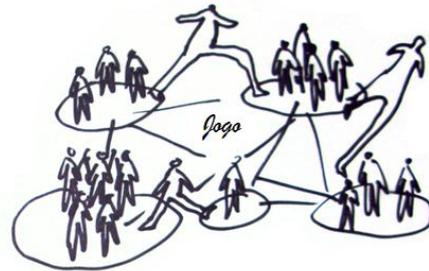
A Figura 33 apresenta a dinâmica de ambientes adocráticos.

DIRECIONADO - HIERARQUIA



BAIXA AUTO-RECONFIGURAÇÃO  
NÃO HÁ MUITO JOGO  
BAIXA INOVAÇÃO  
EXECUTIVO, PRODUTIVO

PLAY-GROUND - ADOCRACIA



ALTA AUTO-RECONFIGURAÇÃO  
MUITO JOGO  
MUITO ATRITO E INOVAÇÃO  
CRIATIVO, DISPERSIVO

Figura 33 -Estruturas Hierárquicas x Estruturas Adocráticas [Mintzberg e McHugh, 1985].

Empresas maduras de alto desempenho conseguem ser flexíveis e adaptáveis como se fossem organizações menores para "ser grande e ainda agir pequeno" ao mesmo tempo. Times são o meio principal de atingir esse estado, pois times auto gerenciados estão rapidamente se movendo em grupos de aprendizagem rápida, flexível, autônomas e têm desenvolvido atitude proativa e sentido de responsabilidade.

Para o MPS ágil, a estrutura ideal é um meio termo proposto por Moe (2011) com o total autogerenciamento da empresa e a proposta que separa por completo a parte de negócios e do time, e isso é relatado por um dos participantes:

*Existe uma diferença na expectativa do cliente e na expectativa do gerente, porque ambas as empresas gostariam de atingir seus objetivos, mas, conceitualmente, a nossa empresa deseja lucrar, enquanto a do nosso cliente economizar. Não teria problema nisso se estivéssemos do mesmo lado, mas ela é quem nos paga. E como time, estamos aqui pra satisfazer todas as vontades do*

*cliente, inclusive as mudanças necessárias. Mas, como funcionários, temos que prestar contas também a alta gestão de nossa empresa. Então, o autogerenciamento não existe por que temos dois chefes com objetivos diferentes. Seria ótimo se eles se alinhassem para que a gente pudesse saber pra onde ir. Por exemplo, duas pessoas foram relocadas do meu time para outro time, sendo que um deles era o nosso product owner que estava em contato todos os dias com o cliente.* Gerente de Projetos empresa .

Esse relato afirma que é impossível o time não ter que responder a uma instância superior, e por isso ele não pode ser completamente auto-gerenciado. Isso já foi identificado por Anderson e McMillan [2003], que apontou que a auto-organização não significa que os trabalhadores, em vez dos gestores, criam o desenho da organização. Não significa deixar as pessoas fazerem o que quiserem. Significa que a gerência se compromete a orientar a evolução dos comportamentos que emergem da interação de agentes independentes, em vez de especificar de antemão o que é um comportamento efetivo.

Esta estrutura de **autocracia** com prioridades dinâmicas na **gestão de portfólio** e **autogerenciamento** não irá funcionar se a **estrutura organizacional** não apoiar que a empresa e seus projetos são **Sistemas Complexos Adaptativos**. E é por isso que esses cinco códigos estão ligados na representação axial dos códigos.

Um sistema adaptativo complexo (SAC) é constituído por um grande número de agentes, cada um dos quais se comporta de acordo com um conjunto de regras. Estas regras exigem que os agentes ajustem seu comportamento ao de outros agentes com os quais eles irão interagir e se adaptar. Um SAC busca identificar características comuns da dinâmica de tais sistemas ou redes em geral [Stacey, 2003].

Um SAC será adaptável se conseguir ajustar as atividades ou reagir aos eventos do ambiente. SAC bem-sucedidos se ajustam de uma maneira que facilita ou permite que o sistema ou projeto alcance o seu objetivo. Por outro lado, um sistema determinístico é um sistema que irá produzir os mesmos resultados se forem adotadas as mesmas condições. O resultado pode ser

comumente previsto se são conhecidas as condições de partida. Um sistema não-linear, ou caótico, pode produzir resultados muito diferentes, mesmo se as condições de partida são quase as mesmas.

Os projetos de software são geralmente sistemas não-lineares. Se nós executarmos um projeto complexo idênticos em três momentos diferentes, eles irão entregar três resultados diferentes. Na abordagem tradicional é suposto que o projeto está inserido em cenários determinísticos, e assim é comum o uso de simulações para desenvolver o resultado mais provável. No entanto, uma pequena mudança, como o calendário de férias de alguém ou uma mudança na data de entrega, pode mudar toda a trajetória de um projeto [Wang e Conboy, 2009].

Considerando que as organizações pesquisadas em sua maioria mantêm seu nível gerencial baseado em abordagens tradicionais, elas deixam de reconhecer que um projeto faz parte de um sistema complexo adaptativo, e conseqüentemente impedem que a organização seja de fato ágil.

### **Códigos relacionados ao código ãValor de Negóciosö**

O código ãvalor de negóciosö está ligado a cinco códigos distintos. A ligação deste com os códigos ãcoachingö e ãescopo abertoö trazem resultados quase nulos, por que ambos são colocados aqui como diferença cultural de ser líder em uma equipe auto-organizável e trabalhar com contratos de escopo aberto.

Todos os códigos ligados à visibilidade junto com a erosão de processo trazem à tona um contexto interessante. Como mostrado anteriormente, os times das empresas entrevistadas não possuem nenhum envolvimento com a parte gerencial e de negócios da organização. Entretanto, estas empresas consideram o time como um conjunto de pessoas que estarão focadas em desenvolver uma solução para um cliente, conforme o relato a seguir:

*ãReuniões de Kick off são engraçadas por que é um dia que nós entramos numa sala e conhecemos no mesmo dia o cliente, os colegas de time, o*

uaer ae equipe, o projeto, as expectativas da empresa, as expectativas do cliente e as restrições do projeto. A reunião sempre termina com a palavra do diretor técnico dizendo: sucesso a todos. Mas, como iremos trabalhar? Qual é o problema do cliente? Por que estamos fazendo aquele projeto? Em caso de problemas como vamos falar com o cliente? Que resultados a companhia espera? A resposta dele é simples: usem o Scrum... ö. Líder de Equipe, empresa

Segundo Moe et al. (2010) não é trivial pedir para que um time se auto organize, pois a transição do trabalho individual para times auto organizados requer reorientação de desenvolvedores e gerentes e toma tempo e recursos, mas, não pode ser negligenciada.

Scrum e outros métodos ágeis não oferecem suporte sobre como implementar liderança compartilhada (*Coach*). Times precisam de tempo para aprender a ser um time para depois aprender a ser um time auto-organizado [Moe et al., 2010].

Como existe a questão de que os **times** dessas empresas não foram preparados para serem **times**, então eles tentam da õmelhor formaõ possível fazer o seu trabalho, e isso normalmente implica em negligenciar o **valor de negócio** do que eles estão desenvolvendo.

Então não há visibilidade nem do **time**, nem da **alta gestão** daquilo que é realizado. Apenas o cliente sabe do valor do seu produto, e este, não é o **valor de negócio** da empresa. Essa falta de **visibilidade** do **valor de negócio** para o **time** faz com que o mesmo apenas foque em seu trabalho técnico e deixe de executar as tarefas da organização, iniciando a **erosão** do processo.

E em alguns casos, quando o time completa a **erosão** e retorna ao processo mínimo, isso também é percebido pelo cliente com o aumento da **dívida técnica** daquele produto, o que diminui o seu valor.

E por fim serão apresentadas as seguintes teorias formuladas baseadas nos dados.

## **T1 ó As organizaçoes pesquisadas não realizam o MPS baseado em métodos ágeis em todos os níveis da organização;**

Era esperado que a iniciativa de MPS baseada em métodos ágeis atingissem também o nível de negócios, entretanto, a alta gestão preferiu adotar para si uma abordagem tradicional de melhoria. Alguns dos questionários apresentaram evidências de que a alta gestão apoiava as adoções de métodos ágeis no nível de time e inclusive apoiou a auto-organização no nível de projeto para a própria divisão do trabalho e para a autonomia da relação com o cliente. Entretanto, a própria alta gestão das companhias entrevistadas preferiram adotar o modelo de otimização em sua forma de trabalho.

Nessas empresas, a estrutura organizacional era hierárquica, e a alta gestão optou por utilizar o programa de MPS como meio de padronizar o modo de produção e melhorar indicadores básicos como produtividade e qualidade. Não há nenhuma preocupação explícita com a possibilidade de mudanças no nível de negócios, e muito menos a preocupação de responder rápido às mudanças.

Então, foi observado que as empresas decidiram, conscientemente ou não, tornar ágil apenas o seu sistema de produção e com isso, ignorar a participação dos membros das equipes na evolução dos negócios. Ou seja, elas agilizaram seu nível operacional e não levaram essa agilidade para os níveis táticos e estratégicos.

## **T2 ó As empresas que agilizam apenas o nível operacional não podem ser consideradas híbridas;**

Segundo Appelo (2010), organizações híbridas são aquelas que podem evitar as desvantagens das equipes funcionais em um ambiente puramente hierárquico e equipes de projetos autônomos em um ambiente puramente em rede. Para que isso aconteça, elas precisam ajustar a visão de negócios dentro de toda a organização.

A partir desta definição, podemos considerar que as empresas pesquisadas não são híbridas, já que elas não flexibilizam sua forma de trabalho,

nem eliminam as desvantagens citadas acima. Elas simplesmente trabalham de maneira diferente nos níveis da organização. O nível operacional trabalha com base em métodos ágeis, enquanto a alta e média gestão trabalha ainda nos moldes tradicionais.

Vale ressaltar que esta forma onde cada camada da pirâmide apresenta um mentalidade diferente não é necessariamente ruim. Isso pode ser vislumbrado em uma empresa que trabalhe em cima de um produto específico que os requisitos do produto mudem muito e os requisitos de negócios mudam pouco. Neste contexto, agilizar apenas o chão de fábrica faz todo o sentido, mas não é essa a realidade das seis empresas que responderam aos questionários, e muito provavelmente, também não a realidade de boa parte das empresas certificadas nestes modelos de qualidade. Normalmente, essas empresas trabalham em diversos projetos que lidam com nichos de mercado completamente diferentes.

Então, podemos inferir que estas empresas não fizeram nenhum tipo de alinhamento dos diversos métodos ágeis ao seu negócio específico e é por isso que o alcance da agilidade se restringe ao chão de fábrica. E se consideramos que a cultura organizacional destas empresas continua sendo de otimização baseada em repetição e controle, elas continuam classificadas como empresas tradicionais ao invés de híbrida.

### **T3 ó As empresas que agilizam apenas seu nível operacional não apoiam a auto-organização.**

Embora estes times tenham autonomia em nível de projeto, eles sofrem interferência relativas a projetos, processo, e até mesmo relativas a negócios. O time a princípio tem total autonomia para decidir como conduzir os projetos e como se relacionar com os clientes. Entretanto, a alta e média gestão interferem nos projetos alterando prioridades, realocando pessoal e recursos, e em casos mais extremos discordando deliberadamente do que era acordado entre time e cliente. Essa situação se mostra desconfortável para o time, que precisa decidir se atende as necessidades do cliente ou se atende as necessidades da alta gestão da organização.

**T4 ó Empresasque agilizam apenas seu nível operacional necessitam de um mecanismo apropriado de gestão de conhecimento para acomodar o seu aprendizado.**

Alguns relatos apontam que, no contexto pesquisado, o aprendizado do time é diferente do aprendizado da empresa. Foi percebido que o time precisa aprender sobre: (i) o produto que estão desenvolvendo, (ii) os negócios do cliente, (iii) os negócios da própria organização, (iv) como evolui o projeto em questão e (v) como se adequar aos processos da organização.

Muitos dos detalhes de tecnologia, projetos e negócio do cliente não são importantes para a evolução da organização. Então, idealmente, o aprendizado da organização não é a soma do aprendizado de todos os times e projetos, mas sim a obtenção do conhecimento necessário para se tornar mais competitiva no mercado. Inclusive, muito desse conhecimento vem de fora da organização conforme é proposto no modelo de inovação aberta. Sendo assim, não se deve confundir conhecimento com competências e experiência.

Competências são muito valiosas para o aprendizado pessoal, mas pouco acrescentam ao conhecimento da organização. Experiência representa a diversidade de situações que foi enfrentada pelos indivíduos, times e empresas que os auxiliam a passar por situações similares, mas não há uma maneira óbvia de se aprender com as falhas e nem de utilizar experiência quando se está diante de um cenário completamente novo. O conhecimento sim pode auxiliar as companhias a tomarem boas decisões em ambientes de constante mudança, já que existe uma base sólida que possibilite fazer previsões de cenários futuros e consequências baseadas nas ações pensadas.

Então as reuniões de pós-morte e coleta de lições aprendidas no projeto podem auxiliar os indivíduos a espalhar sua experiência, mas isso se torna limitado, de certa forma, quando falamos de conhecimento. Assim, se mostra necessária a criação de um novo modelo de gestão do conhecimento nesse tipo de cenário.

### **T5 - Os elementos norteadores dos modelos de qualidade refletem a cultura da organização;**

Foi relatado que os elementos que afetam as práticas de um determinado processo, práticas específicas no CMMI e Resultados Esperados do Processo no MPS.BR, representam o modo de como o nível operacional trabalha. Já os elementos que afetam o trabalho de todos os envolvidos no processo, práticas genéricas do CMMI e Atributos de Processo do MPS.BR, espelham a cultura da organização. Mostra qual é a sua forma de trabalho, estrutura organizacional e modelo de negócios.

Essa descoberta surgiu a partir dos relatos dos times que começaram a sentir dificuldades em estar aderentes a esses elementos durante as iniciativas de MPS. Ressaltando que essas práticas foram aquelas que mais sofreram com a erosão de processo após a obtenção da certificação.

### **T6 - Negócio é a base para a MPS ágil;**

O fato de o nível operacional utilizar métodos ágeis não esconde a preferência da alta gestão pelo controle. As empresas passam a controlar, medir e otimizar todas as atividades de times ágeis, e assim impedem que estes mesmos times sejam partes ativas no processo de adaptação aos negócios.

Com isso, a adaptação das empresas não passa pelo ajuste da estrutura organizacional de mudança de comportamento estratégico diante de mudanças, e o que acontece, na verdade, é que os times precisam se adaptar a nova realidade. Isso implica em, quando houver mudanças nos negócios, ao invés da organização identificar a melhor forma de solução, repassarão aos times novas exigências sobre como eles irão trabalhar (interrupção), para que este se molde a nova realidade de negócios. Então MPS ágil é aquele que não só prepara a organização para mudanças, mas que alinha o time aos negócios da organização.

### **T7 - Modelos de qualidade e métodos ágeis apresentam choque de cultura, principalmente se adotados em conjunto;**

Esta teoria retoma as primeiras discussões teóricas sobre a junção de métodos

ágeis e modelos de qualidade. Lá foram levantadas as primeiras incompatibilidades no modo de trabalho que atrapalhariam o trabalho conjunto. Algumas das incompatibilidades levantadas foram: o excesso de documentação, ênfase no processo e excesso de atividades que não geram código executável [Boehm e Turner, 2002].

Mas, estas discussões teóricas que partilham da opinião de incompatibilidade entre as abordagens consideram apenas o nível de time e não de negócios. Ao analisar os trabalhos que defendem esta junção percebemos que existe uma clara divisão entre o que deve ser feito (dirigido pelos modelos de maturidade) e o como deve ser feito (auxílio das abordagens ágeis). Este fato apresenta uma conciliação entre o seu uso em conjunto.

Quando elevamos a mentalidade para nível de negócios percebemos um verdadeiro desafio. Os modelos de maturidade necessitam de certa estabilidade para que o processo possa ser repetido o suficiente, para que as empresas amadureçam no uso do processo. Os métodos ágeis incentivam as mudanças para que as companhias se adaptem ao ambiente e tenham esta adaptabilidade como vantagem competitiva.

Este desafio é diferente daquele proposto por organizações híbridas que tentam flexibilizar o processo em um meio termo entre ágil e tradicionais, por que no nível de negócios a decisão a ser tomada é: o que otimizar? E o que adaptar? Nas empresas pesquisadas, quatro decidiram não levar os métodos ágeis ao nível de negócio, enquanto as outras duas que decidiram levar os métodos ágeis a este nível decidiram por não se certificar. Esse paradoxo é mostrado nos relatos abaixo:

*Tomamos a decisão de utilizar o CMMI para aquisição, para nos auxiliar no acompanhamento dos contratos desta organização. Iniciamos esta implementação com a abordagem scrum onde subcontratávamos um time. Entretanto, percebemos que cada contrato diferente exigia um levantamento diferente, um acompanhamento diferente, uma forma diferente de recebimento, por que envolvia clientes diferentes, negócios diferentes, fornecedores diferentes em contextos diferentes. O nosso único elemento estável era o processo*

licitatório que já é guiado pela lei 8.666. Então em um determinado momento percebemos que o CMMI para aquisição e o Modelo de Aquisição do MPS.BR não seria útil, por que não temos como objetivo contratar melhor, pois o processo licitatório não nos permite isso, então descobrimos que o CMMI.ACQ não se apresentou o modelo mais adequado para o negócio da organização. Ainda hoje mantivemos alguns hábitos ágeis, como envolver o time em todas as decisões de projeto e de negócio. Temos autonomia de negócios para elaborar junto ao governo políticas públicas, como a utilização da Instrução Normativa 04, pois os métodos ágeis nos obrigaram a discutir sobre como os negócios evoluem e como conciliar os interesses políticos ao dever de servir ao cidadão. Gerente de Unidade, Empresa .

Quando decidimos pela iniciativa de MPS estávamos num momento de expansão que começamos a explorar novos nichos de mercado. A implementação dos modelos de maturidade nos auxiliaram muito a refletir sobre o que deveríamos fazer para organizar o nosso desenvolvimento, mas começamos a ter dificuldades em adaptar o processo ao novo negócio. Sempre tínhamos que montar uma equipe nova, com um cliente novo e uma tecnologia nova, o processo sofria muitas mudanças ao longo do tempo, então o nosso processo nunca ficava estável para iniciarmos as coletas para o MPS.BR e de repente nos damos com 4 ou 5 processos diferentes, com passos e artefatos completamente diferentes.

Resolvemos zerar tudo e adotar o Scrum, e ficamos de ajustar o processo ao longo do uso. Para a nossa surpresa com 2 a 3 meses estávamos com os projetos alinhados e os processos não eram tão diferentes um dos outros. Então decidimos levar este tipo de atitude para alta gestão, e hoje tudo é automatizado, gerado de acordo com base histórica e sempre que possível alguns dos membros dos times colaboram com as novas decisões da organização. Inclusive é comum pessoas transitarem entre os projetos. Consideramos hoje voltar a utilizar os modelos de maturidade por que precisamos controlar os nossos projetos que já estão estáveis. Gerente de Projetos, empresa .

Percebemos pelos relatos que existe essa incompatibilidade entre as abordagens no nível dos negócios, e uma das organizações até propõe uma proposta de utilização híbrida em que alguns projetos que possuem uma volatilidade menor de negócios sejam gerenciados pelos modelos de maturidade.

#### 4.4. SUMÁRIO DO CAPÍTULO

Este capítulo apresenta a análise de dados referentes a cada uma das questões de pesquisa. Para a questão de pesquisa QP1 foi observado que seis das oito (75%) das empresas pesquisadas conseguiram as certificações CMMI e MPS.BR utilizando uma abordagem iniciada por métodos ágeis para melhoria de processo de software. Há um indício de que esta abordagem possa apresentar indicadores de sucesso maiores do que outras iniciativas de MPS tais como *six sigma* e CMM.

Para a questão de pesquisa QP2 é observado que esta abordagem de MPs baseada em métodos ágeis não apresentam vantagens nem desvantagens em relação as implementações do MPS.BR quando consideramos tempo de implementação, retorno de investimento e custo médio. Entretanto, vale ressaltar que as empresas pesquisadas nesta tese também passaram por avaliações CMMI e há um impacto no custo final da iniciativa, mas na coleta de dados não havia a separação dos valores da avaliação.

Ainda para a QP2 foram levantados os indicadores qualidade, produtividade, faturamento, tempo médio e custo médio. Devido ao pequeno número de empresas participantes é possível generalizar o resultado devido a uma insegurança estatística. A título de curiosidade, os indicadores das empresas pesquisadas foi comparado aos mesmos indicadores das empresas pesquisadas no iMPS há indícios de que existe um melhor resultado no quesito de qualidade e produtividade, e piora no faturamento, tempo médio e custo médio. Contudo, estes resultados só serão conclusivos a partir dos dados de um maior número de empresas reais.

Ainda na QP2 foi verificado que as empresas pesquisadas sofriam uma

maior erosão dos processos CMMI/MPS.BR maiores do que aquelas verificadas ao processo Scrum. Mesmo dentro dos modelos de qualidade, se foi percebido que a erosão referente aos elementos norteadores (práticas genéricas do CMMI e atributos de processo do MPS.BR) do que nas práticas referentes a um processo específico (práticas específicas do CMMI e resultados esperados do MPS.BR). Isso pode significar que o time não entende o valor de negócio das práticas adotadas pela organização (mais evidentes nos modelos de qualidade) do que no seu trabalho diário (mais evidente no Scrum).

Para a QP3 foram utilizadas as entrevistas abertas para se verificar os principais elementos de uma melhoria de processo de software ágil. Alguns códigos emergiram destas pesquisas e o principal deles foi o código ãNegócioö o que implica que a melhoria de processo de software ágil é mais percebida no nível de negócios da organização. Outros códigos emergiram ao longo da pesquisa e a relação entre estes códigos foi apresentada na representação coaxial proposta. Ao fim desta análise baseada na teoria fundamentada em dados foram propostas sete teorias sobre a adoção de MPS baseado em métodos ágeis e MPS ágil, que nesta pesquisa se mostraram como elementos diferentes, ou seja o MPS baseado em métodos ágeis utilizados por estas empresas não se encaixa na denominação de MPS ágil definida para esta tese.

## 5. Conclusão

### 5.1. DESCOBERTAS

Para facilitar a localização das contribuições desta tese será colocado aqui um espaço para as descobertas.

#### *5.1.1.EMPRESAS DE SOFTWARE PODEM OBTER NÍVEIS DE MATURIDADE EM MODELOS DE QUALIDADE UTILIZANDO ABORDAGENS PARA MELHORIA DE PROCESSO BASEADO EM MÉTODOS ÁGEIS.*

Esta descoberta é a resposta da questão de pesquisa QP1. Empresas de software tem conseguido obter certificações CMMI e MPS.BR adotando abordagens de melhoria de processo baseadas em métodos ágeis. Este resultado seria inédito por si só, caso estas empresas conseguissem levar a mentalidade ágil também para o nível gerencial. Entretanto, o que chamamos de mentalidade ágil que é desenvolvida inicialmente segue até que o nível operacional (time), esteja usando adequadamente as abordagens ágeis. Então o que temos observado é que estas organizações apenas antecipam a adoção de um método ágil e não adotam tais métodos baseados na mentalidade ágil.

#### *5.1.2.QUAIS OS BENEFÍCIOS OBSERVADOS POR ESTAS ORGANIZAÇÕES AO SE ADOTAR ESTA FORMA DE MELHORIA DE PROCESSO? E EM LONGO PRAZO?*

Ainda que esta pesquisa não traga resultados conclusivos devido ao pequeno número de empresas que responderam ao questionário, são mostrados indícios de que realizar programas de MPS baseados em métodos ágeis, quando comparadas com empresas pesquisadas no iMPS, não traz nenhum benefício quando

observados o tempo gasto e o custo da implementação, assim como o retorno de investimento. O benefício percebido é que percentualmente um número maior de empresas que adotaram o MPS baseadas em métodos ágeis aumentaram a produtividade e a qualidade, se comparadas com as empresas pesquisadas pelo iMPS. Por outro lado, o faturamento, o custo médio por projeto e o tempo médio por projeto apresentam um desempenho pior.

Se compararmos a taxa de sucesso com outros programas formais podemos ver que esta abordagem apresenta 75% de implementações bem sucedidas, o que é melhor do que as outras iniciativas pesquisadas. Observando estas organizações em longo prazo, é percebida a erosão de processo. Esta erosão se apresenta maior nas práticas relacionadas aos modelos de qualidade do que nas práticas relacionadas aos métodos ágeis.

Este resultado por si só é inédito quando tratamos de empresas que adotam estes métodos ágeis e responde parcialmente a QP2. Esta descoberta auxilia a preencher as lacunas de pesquisas L6 [Salo, 2007] ao mostrar como ocorre a melhoria nestas organizações, nesta pesquisa como ocorre a erosão dos processos e evolução dos indicadores. A lacuna de pesquisa L9 [Pikkarainen, 2008] sugere que é necessário pesquisar a assimilação das práticas ágeis nestas empresas CMMI, e para esta descoberta em particular, temos os resultados do *Nokia Test* (questionário sobre a adoção do Scrum) indicando como as empresas estão adotando o Scrum nas organizações.

### *5.1.3. COMO A MELHORIA DE PROCESSO ÁGIL ÷INDUTIVA÷ ESTÁ RELACIONADA À MELHORIA DE PROCESSO ÷PRESCRITIVA÷ ADOTADA PELO CMMI E MPS.BR?*

O MPS ágil está presente no nível operacional da organização (equipe de desenvolvimento). Entretanto, de uma maneira geral, a organização é dividida em duas partes distintas que são: o nível operacional já citado e o nível gerencial. O nível gerencial prefere adotar uma política de otimização baseado em processos repetitivos e que se encaixam bem na visão do CMMI, deixando a

parte de inovação e de apoio a mudanças para a equipe de desenvolvimento.

Embora o time utilize de fato abordagens ágeis, tais como incremento de processos e melhoria adaptativa, eles não encontram apoio para se manter neste estado de autonomia e não são solicitados pela alta gestão quando há mudanças do nível de negócio, e assim se tornam passivos diante das mudanças necessárias.

Então podemos dizer que o MPS ágil, nas empresas pesquisadas, foi considerado como fazer na estrutura do MPS prescritivo (O que fazer). Isto não as diferencia das empresas que utilizam abordagens ágeis dentro de seu processo apenas como ferramentas.

Novamente, esta descoberta é considerada inédita. Uma observação interessante é que as empresas que levaram essa forma de MPS para o nível gerencial não levaram adiante o processo de certificação. A grande novidade nesta questão é que, do ponto de vista de MPS, não existe diferença entre as empresas que usam metodologias ágeis, como abordagem inicial de MPS, para aquelas que simplesmente adotam uma abordagem ágil, como Scrum ou XP, dentro do seu processo. Isso poderia ser diferente, caso as empresas certificadas levassem isso a um nível de negócio.

Esta descoberta responde parcialmente a QP3 e auxilia a preencher as lacunas de pesquisa L5 e L6 [Salo, 2007] referentes à melhoria contínua e a integração das duas abordagens de MPS em questão.

#### *5.1.4. NEGÓCIO É A BASE PARA A MELHORIA DE PROCESSO ÁGIL?*

Toda a discussão desta descoberta já foi discutida na teoria T6, ficando para esta parte ressaltar que este ponto responde parcialmente a questão QP2 e auxilia no preenchimento da lacuna L1 [Anderson e Selles, 2008]. Anderson e Selles propõem a criação de um arcabouço para criar, adaptar e modificar o uso de abordagens ágeis nas empresas. Entretanto, esta pesquisa sugere que métodos ágeis são amplamente usados em um contexto de mudanças, então esse arcabouço só teria seu uso aproveitado em empresas que não possuem uma

cultura de agilidade, como as que foram pesquisadas aqui.

Outra lacuna de pesquisa que é complementada por esta pesquisa é a L2 [Moe, 2011], que aponta que o autogerenciamento deveria acontecer em todos os níveis da organização para apoiar a melhoria de processo de software ágil. Nesta pesquisa, é observado que a alta gestão deve avaliar suas necessidades de mudança para adotar uma melhoria de processo que privilegie a adaptação ao invés da otimização.

Outras lacunas de pesquisa que são complementadas por esta descoberta são a L5 e L6 [Salo, 2007] e L9 [Pikkarainen, 2008], por adicionar a visão de negócios e a importância da alta gestão ao que já foi visto nas outras descobertas.

#### *5.1.5. AS ORGANIZAÇÕES ÁGEIS+CMMI OU ÁGEIS+MPS.BR NÃO SÃO HÍBRIDAS?*

A discussão desta descoberta já foi feita na teoria T2 ficando para esta parte ressaltar que esta descoberta responde parcialmente a QP3 e auxilia no preenchimento da lacuna de pesquisa L1 [Anderson e Selles, 2008] delimitando em quais empresas a sugestão de um arcabouço é válida. A lacuna L2 [Moe, 2011] também é complementada, porque é identificado quais são as empresas inovadoras, denominadas por Moe como guiadas por mudanças, e as empresas *otimizadoras*.

O que define essa característica não é o completo autogerenciamento das equipes de projeto, mas sim o alinhamento de negócios às necessidades de mudanças definidas pela alta gestão, e dessa forma, a adocracia (e não o autogerenciamento) é uma consequência e não uma causa. As lacunas L5 e L6 [Salo, 2011] e L9 [Pikkarainen, 2008] também são atingidas por esta descoberta, já que as empresas pesquisadas não são híbridas e esse estado parece ser o mais comum em empresas que adotam métodos ágeis, e se certificam em modelos de qualidade.

#### *5.1.6. MODELOS DE QUALIDADE E ABORDAGENS ÁGEIS APRESENTAM CHOQUE DE CULTURA, PRINCIPALMENTE SE ADOTADOS EM CONJUNTO.*

Esta descoberta já foi discutida na teoria T7 formulada no capítulo anterior, restando aqui apontar que a mesma responde parcialmente a QP3 e auxilia o preenchimento das lacunas de pesquisa L5, L6 [Salo, 2009] e L9 [Pikkarainen, 2008] explicitando o choque de cultura entre ambas as abordagens.

#### *5.1.7. OS TIMES NESTAS ORGANIZAÇÕES NÃO SÃO AUTO-GERENCIADOS;*

A discussão dessa descoberta já foi realizada na teoria T3, proposta no capítulo anterior, e essa responde parcialmente a QP3 e complementa os resultados de Moe (2011) sobre a lacuna de pesquisa L2, que trata do completo autogerenciamento ao afirmar que não há autogerenciamento mesmo nas equipes uma vez que elas sofrem interrupções da alta-gestão. As lacunas de pesquisa L5, L6 e L9 também são complementadas.

#### *5.1.8. O APRENDIZADO NO MPS ÁGIL NECESSITA DE MECANISMOS DIFERENTES DAQUELES PROVIDOS NO MPS TRADICIONAL.*

A discussão dessa descoberta já foi feita na teoria T4 levantada no capítulo anterior. Esta descoberta responde parcialmente a questão QP3 e corrobora com a lacuna de pesquisa L4 [Salo, 2007] indicando que realmente é necessário identificar novos mecanismos de aprendizado neste tipo de ambiente que envolve métodos ágeis.

#### *5.1.9.A CULTURA ORGANIZACIONAL É PERCEBIDA NO CMMI E MPS.BR A PARTIR DE PRÁTICAS GENÉRICAS E ATIBUTOS DOS PROCESSOS RESPECTIVAMENTE.*

Esta descoberta já foi discutida na teoria T5 proposta no capítulo anterior. Esta descoberta responde parcialmente a questão QP2 e auxilia na resolução das lacunas de pesquisa L10 [Santana et al., 2009] que propõe o desconhecimento do papel dos elementos norteadores no relacionamento entre métodos ágeis e CMMI.

#### *5.1.10. A MELHORIA ADAPTATIVA DE PROCESSO ACONTECE NO NÍVEL DE TIME E NÃO DE ORGANIZAÇÃO;*

Era esperado que, em uma iniciativa de MPS ágil, as melhorias de processo adaptativas fossem tomadas no nível de negócios, mas a alta gestão preferiu adotar uma abordagem tradicional de melhoria no seu nível. Sem esse comportamento da alta gestão, era esperado que a melhoria adaptativa não acontecesse nas organizações. Entretanto, os times realizaram a melhoria adaptativa baseada no apoio ferramental (Automatização) e nos incrementos de processo realizados nas retrospectivas ao fim de cada iteração.

Esta descoberta responde parcialmente a questão QP3 e auxilia na resolução das seguintes lacunas de pesquisa L5 e L6 [Salo, 2007].

#### *5.1.11. A EROSIÃO DO PROCESSO SE DÁ POR QUE MUITOS DOS TIMES NÃO ESTÃO ALINHADOS COM A CULTURA DA ORGANIZAÇÃO OU EXISTE A FALTA DA VISIBILIDADE DO VALOR DE NEGÓCIO DO PROCESSO PARA O TIME;*

Foi identificado que a erosão do processo foi maior nos elementos norteadores do que nas práticas do Scrum. Este fato indica que o valor dos elementos dos modelos de qualidade é percebido pela organização, e não pelo time. Por isso,

são abandonados primeiro e em maior grau. Este abandono pode ser tanto devido à falta de visibilidade no valor de negócio de determinadas práticas para os times, quanto pela própria organização que cria o processo para que este se adapte ao modelo de maturidade e após a certificação abandona aquele processo. O processo mínimo resultante depois da erosão deste é mais parecido com o processo Scrum da organização do que aquele aderente aos modelos de qualidade.

Podemos considerar que há aqui a questão da falta do alinhamento de negócios entre equipe e alta gestão sobre o porquê da adoção dos modelos de qualidade. Sem essa consciência da importância do processo adequado a um modelo de qualidade, a equipe então abandona o processo. Normalmente, este abandono se dá na transição de projetos mais complexos para aqueles mais simples ou em momentos de restrições de prazo.

Neste momento, a equipe abandona determinadas práticas e não as retomam posteriormente. A equipe é forçada a retomar certas práticas quando as auditorias de qualidade detectam a quebra do processo. Quando uma prática é abandonada de forma recorrente, as organizações tem a tendência de ignorar o sentimento da equipe e questionar o porquê daquele abandono, normalmente é sugerido que a prática seja retomada. Pode também acontecer que a prática abandonada seja ignorada pela própria organização, o que indica que não há sequer o alinhamento dos negócios com a necessidade da utilização de um modelo de qualidade.

Esta descoberta responde parcialmente a questão QP2 e auxilia a na resolução das seguintes lacunas de pesquisa L2, L5, L6 e L9.

#### *5.1.12. CODIFICAÇÃO COAXIAL DOS FATORES ENVOLVIDOS NA JUNÇÃO DE METODOLOGIAS ÁGEIS E MODELOS DE QUALIDADE.*

A codificação coaxial mostrada na Figura 18 apresenta a primeira visão de relacionamentos entre elementos que estão presentes na junção de Metodologias

Ageis e Modelos de Qualidade, este elemento pode auxiliar na discussão sobre a junção destas abordagens em um mesmo ambiente.

Esta descoberta responde parcialmente a questão QP3 e complementam informações sobre a resolução das seguintes lacunas de pesquisa, L5 [Salo, 2007] e L9 [Pikkarainen, 2008] .

## 5.2. DISCUSSÃO DOS RESULTADOS

Uma parte da discussão dos resultados já foi feita nas seções descobertas (Seção 5.1) e teorias (Seção 4.3). Mas ainda se faz necessário debater sobre alguns dos pontos presentes nesta tese.

O primeiro deles é a frustração de não encontrar nenhuma empresa certificada que levou o MPS ágil para o nível de negócios dentro da organização. Não foi possível avaliar como se dá o MPS ágil em empresas certificadas em modelos de maturidade, mas este fator revelou a importância da cultura da organização para tomar a decisão de tornar ágil o seu modelo de negócios como um todo.

O que vimos foi que as empresas que se certificaram preferiram manter seu modelo de negócios focados na otimização e na repetição do contexto, enquanto as empresas que perceberam as mudanças em seu ambiente de negócios preferiram agilizar o seu modelo de negócio e descobriram que não precisavam do modelo de maturidade. Uma delas até considera utilizar os modelos de maturidade em projetos mais estáveis.

Apesar desta diferença conceitual parecer o atributo de maior impacto nesta escolha, devemos levar em consideração outras coisas. O próprio nível dois do CMMI e nível F do MPS.BR tem o foco em projetos, e talvez as organizações não precisaram alinhar de fato os seus negócios aos times e consequentemente alinhar a cultura organizacional à agilidade apresentada no seu chão de fábrica. Talvez com a ida para Nível 3 do CMMI e Nível E do MPS.BR isso possa ser avaliado.

Outro fator que pode ter influenciado ainda é a falta de técnicas

anunciadas sobre como as abordagens ágeis podem influenciar o negócio de uma empresa. A princípio estas empresas são ãnovatasö na adoção de métodos ágeis e se considerarmos quem são as empresas ágeis de fato, conforme apresentado na Figura 5, as empresas pesquisadas aqui estão na categoria das preguiçosas e não tem realmente a mentalidade ágil. Isso implica que possivelmente elas só irão subir da agilidade para o nível de negócios quando houver um modelo bem definido para isso. Se considerarmos a teoria de Redwine (1984 e 1985), isso só aconteceria daqui a uns 10 a 15 anos, já que essa questão de agilidade nos negócios começou a ser discutida em 2010. A utilização de métodos ágeis junto com modelos de maturidade começou a ser discutida em 2001, portanto há 11 anos, e já existem algumas coisas bem difundidas que foram trazidas a estas organizações através das suas consultorias.

O fato dessas empresas não serem híbridas não é necessariamente ruim, já que em alguns modelos de negócios existem mais mudanças na concepção e construção do produto do que nos negócios em si. E ainda, pode ser melhor deixar a agilidade restrita ao nível operacional, enquanto o nível de controle gerencial é baseado numa estratégia de otimização. Os resultados apresentados para a QP2 trazem bons resultados na questão de custo e taxa de sucesso da adoção dos programas. Logo, o fato de estas empresas não se diferenciarem, do ponto de vista de cultura, daquelas que implementam métodos ágeis no meio ou após a adoção de obter as certificações de qualidade, parece se beneficiar da adoção antecipada destes métodos.

Outro fator que vale a pena ser comentado é o resultado inédito da observação da erosão de processo neste tipo de ambiente. Este tópico também possui diversas lacunas de pesquisa, tanto que a primeira identificação de um modelo de processo que contemplava a erosão de processo surgiu em 2008 [Coleman e Connor, 2008]. Este foi o primeiro estudo envolvendo erosão de processo em empresas brasileiras e como esta erosão ocorre em um determinado ambiente. Já era sabido informalmente que as empresas abandonavam algumas práticas dos modelos de maturidade após a certificação por que elas consideravam estas práticas desnecessárias. Esta formalização sobre como a

erosão de processo ocorre nas organizações pode ser útil para as empresas de consultorias auxiliarem as organizações a alinhar o que é feito no CMMI e MPS.BR.

Outro ponto importante de se discutir é A Melhoria Adaptativa de Processo, que foi formalizada como técnica em 2010 na conferência SPICE e se mostra uma área de pesquisa ainda em formação. A suposição inicial para esta pesquisa é que essa melhoria fosse apoiada pela alta gestão por se tratar de uma adaptação aos negócios, entretanto foi percebido que foi o time quem exerceu essa melhoria adaptativa no ambiente interno. Ao fim de cada iteração durante as reuniões de retrospectiva, o time se reunia também para discutir sobre como melhorar o seu comportamento diante das novas exigências.

Outro ponto a ser discutido é que, ao observar os indicadores das empresas avaliadas MPS.BR a partir do relatório iMPS [Travassos e Kalinowski, 2008 a 2011] temos a sensação de que a maioria das empresas que obtêm a certificação MPS.BR se tornam mais caras, já que há o aumento do faturamento e diminuição da produtividade, demoram mais a entregar os projetos e não apresentam um aumento significativo de qualidade. Este quadro não é alterado de forma significativa quando as empresas adotam métodos ágeis para iniciar os programas de MPS.

Considerando que não houve mudanças significativas nos indicadores organizacionais nas empresas certificadas que utilizaram o MPS baseado em métodos ágeis, há a indicação que os métodos ágeis não auxiliam nem atrapalham na obtenção da certificação de qualidade. Métodos ágeis também não podem ser considerados um fator que interfere na manutenção do processo concebido para se adequar ao modelo de qualidade.

Por último, vale comentar a respeito da utilização de estudos de caso e teoria fundamentada como métodos de pesquisa. A área de métodos ágeis ainda é uma área em que, em alguns pontos, a academia está alguns passos atrás da indústria o que significa que muito do que a indústria realiza, seja bom ou ruim, ainda não foi rigorosamente investigado por um olhar científico. Esta realidade propicia que o melhor local para se fazer pesquisa sobre a adoção de métodos

ageis e a propria inuustria (indústria como laboratório). A teoria fundamentada mostra extrema relevância para apresentar características qualitativas, como por exemplo, a codificação axial que apresenta como estão relacionados as características da pesquisa através dos códigos. Este quadro também é inovador do ponto de vista da pesquisa sobre a adoção de métodos ágeis em ambientes CMMI.

### 5.3. LIMITAÇÕES E AMEAÇAS À VALIDADE

Algumas limitações e ameaças à validade puderam ser observadas no estudo mesmo com todos os cuidados e atenuações que foram promovidos pelos pesquisadores.

- **Utilização do Estudo de Caso como método de pesquisa:** Como já citado no Capítulo 3 (Metodologia), estudos de caso são conhecidos por possuir uma fraca capacidade de generalização; isto foi atenuado com a aplicação de estudos de caso em mais de um local, e por não ser efetivo sem um protocolo ou método adequado. Neste estudo usamos o protocolo de Brereton et al. (2008), pois devido ao caráter *in vivo* das questões de pesquisa este método pareceu o mais adequado, embora que estas razões não eliminem as ameaças do método.
- **Amostra pequena de Empresas Pesquisadas:** Este estudo de caso contou com participação de seis empresas, das quais, quatro conseguiram as certificações de qualidade e duas não conseguiram. Entretanto, a população máxima de empresas que poderiam se encaixar no estudo são 39 Empresas (42 empresas que possuem CMMI e MPS.BR ó 3 Empresas que foram eliminadas do estudo). Sendo assim, a cobertura da pesquisa foi de no mínimo de 10%, que já é considerável quando vamos para a estatística. Contudo, acreditamos que essa relação percentual seja maior já que existem outras empresas que não usam métodos

ageis. Este número de empresas pode influenciar na generalização dos resultados finais, por isso alguns indicadores como, Retorno do Investimento e Tempo Médio de Projetos foram desconsiderados e só foram comentados aqueles que apresentaram resultados muito diferentes dos utilizados pelo iMPS. Mesmo que a generalização não seja possível, estes dados tem a validade de serem os primeiros resultados apresentados nesta estrutura e podem ser complementados com outros estudos, e posteriormente aglutinados em estudos secundários.

- **Utilização do iMPS como conjunto de empresas controle para comparação:** A maior restrição ao uso do iMPS é que o mesmo considera todas as empresas certificadas no MPS.BR independentemente de elas conseguirem ou não o CMMI. Então empresas MPS.BR + CMMI podem apresentar um comportamento diferente daquelas que são certificadas apenas no MPS.BR. Entretanto, acredita-se que este fator não tenha relevância significativa, por que não há muita diferença entre os dois modelos. Inclusive, no guia geral do MPS.BR [MPS.BR, 2011a] há uma seção mostrando as poucas adaptações necessárias para uma empresa CMMI se certificar MPS.BR, e já há casos em que a avaliação nos dois modelos foi conjunta [Rocha et. Al., 2009] Portanto, acreditamos que o fato das empresas pesquisadas utilizarem métodos ágeis teve um impacto muito mais significativo do que o CMMI.
- **Utilização de questionários (*survey*):** Também, como foi explicado no Capítulo 3 (Metodologia), a utilização de *survey* depende exclusivamente da boa fé e da total capacidade dos respondentes, e estes problemas foram atenuados a partir de correspondências eletrônicas explicativas, encontros face a face quando possível, e a adoção de um protocolo para a condução do *survey*, objetivando minimizar os problemas.

- **Utilização da Técnica Teoria Fundamentada para Análise Qualitativa:** A utilização de teoria fundamentada em dados pode trazer um viés a partir da característica interpretativa dada pelo pesquisador. A atenuação deste viés foi dada a partir da utilização de uma nuvem de palavras para identificar de fato os códigos mais relevantes. Todavia, alguns dos resultados qualitativos relevantes só foram descobertos a partir da utilização desta técnica.

#### 5.4. IMPLICAÇÕES PRÁTICAS E TEÓRICAS

A principal implicação teórica para esta pesquisa é a descoberta da importância de que a mentalidade ágil, também na melhoria de processo, está voltada para negócios. Cada vez mais pesquisas põem a ênfase dos métodos ágeis na parte de negócios e menos na parte técnica. Isso implica que, daqui em diante, pesquisas sobre melhoria de processo de software baseadas em métodos ágeis também devem considerar esta visão. Foram apresentados aqui resultados sobre erosão de processo em métodos ágeis e como a melhoria adaptativa de software é implementada nestes ambientes. Estes resultados não são conclusivos, mas, servem como base para futuros estudos.

As implicações práticas apontam para que a adoção de abordagens ágeis de MPS sejam mais baratas e mais bem sucedidas do que o normal quando se trata de certificações MPS.BR. Outra implicação prática é útil para as empresas que implementam CMMI e MPS.BR é que se deve dar atenção está as práticas genéricas (CMMI) e atributos de processo (MPS.BR). Esses elementos norteadores devem ser concebidos por atividades que gerem algum tipo de valor para a organização e que este valor seja também compartilhado com o time para evitar a erosão do processo CMMI.

## 5.5. TRABALHOS FUTUROS

Alguns trabalhos futuros para esta pesquisa são derivados das lacunas de pesquisas que não foram totalmente preenchidas, enquanto outros são novas lacunas abertas por esta pesquisa.

**TF1** se refere à lacuna de pesquisa L1 [Anderson e Selles, 2008], pois não foi proposto um arcabouço que possa ser usado para criar, modificar e adaptar situações específicas em metodologias ágeis. Apenas, foi identificado que este arcabouço será plenamente utilizado por empresas que não tem a cultura ágil, que são as empresas participantes desta pesquisa.

**TF2** relativa à lacuna de pesquisa L2 [Moe, 2011] que trata do completo autogerenciamento. Este resultado evolui, em alguns aspectos, os resultados encontrados por Moe, pois nesta pesquisa acredita-se que a melhor organização de pessoas na organização seria uma estrutura adocrática ao invés do completo autogerenciamento.

**TF3** se refere à lacuna de pesquisa L3 que trata dos métodos ágeis em ambientes globais. Esta lacuna que não foi evoluída nesta pesquisa, se mantendo no mesmo estado daquele definido na introdução.

**TF4** se refere à lacuna de pesquisa L4 onde Salo (2007) levanta a necessidade de identificar os mecanismos de aprendizado e gestão de conhecimento em ambientes onde a melhoria do comportamento do time é o núcleo do MPS. Neste trabalho foi apresentado que este tipo de conhecimento (experiência e competência técnica) é pouco valorizado para a melhoria de processo organizacional. Porém, ainda está em aberto a definição de um mecanismo de aprendizagem neste tipo de ambiente.

**TF5** se refere a lacuna de pesquisa L7 onde Dingsoyr et. al. (2012) afirmam que é preciso fornecer uma compreensão do processo de aprendizagem que ocorre dentro das equipes de desenvolvimento de software ágeis ao se conduzir reflexões iterativas para a melhoria de processo de software. Esta pesquisa mostrou que esse aprendizado iterativo também precisa considerar aspectos de negócio.



*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

**TF05** refere a lacuna de pesquisa L8 que aponta que pesquisas devem ser realizadas em níveis de maturidade mais altos do CMMI e MPS.BR. Esta lacuna de pesquisa se mantém, já que aqui só foram pesquisadas empresas com o nível de maturidade dois (CMMI) e F (MPS.BR)

**TF76** Outro trabalho futuro sugerido seria identificar como ocorre o MPS ágil em empresas que adotam essa melhoria de processo de software indutiva aos níveis de alta gestão. Esse era o objetivo original desta pesquisa, entretanto, foi observado que as empresas adotam métodos ágeis para o nível operacional.

## Referências

- [Allison, 2005] Allison, I. (2005). Towards an Agile Approach to Software Process Improvement: Addressing the Changing Needs of Software Products. In Communications of IIMA; vol. 30, pp. 67-76.
- [Allison & Merali, 2007] Allison, I. & Merali Y. (2007). Software process improvement as emergent change: A structural analysis. Information and Software Technology 49: pp - 668-681.
- [Ambler, 2009] Ambler, S. (2009). Answering the "Where is the Proof That Agile Methods Work" Question. Available in: <http://www.agilemodeling.com/essays/proof.htm>
- [Anderson, 2005] Andersson, D.J. (2005). Stretching Agile to Fit CMMI Level 3. In: Agile Development, Denver.
- [Anderson e McMillan, 2003] Anderson, C. e McMillan, E. (2003). Of ants and men: Self-organized teams in human and insect organizations. Emergence 5. 29-41
- [Antonioni, 2010] Antonioni, J. (2010). Programa MPS.BR - Avanços, conquistas e resultados alcançados. Wamps 2010.
- [Appelo, 2010] Appelo, J. (2010). Management 3.0. Addison Wesley.
- [Baker, 2005] Baker, S. W. (2005). Formalizing Agility: An Agile Organization's Journey toward CMMI Accreditation. In proceedings of Agile Development Conference.
- [Baker, 2006] Baker, S. W. (2006) Formalizing Agility, Part 2: How an Agile Organization Embraced the CMMI. Agile 2006 Conference, Minneapolis, Minnesota.
- [Baskerville et al, 2010] Baskerville, R., Pries-Heje, J., e Madsen, S. (2010). From exotic to mainstream: A 10-year odyssey from internet speed to boundary spanning with Scrum, in: agile software development: current research and future directions, (eds.) Dybå T., Dingsøyr T., Moe, N.B., Springer, 2010, Chapter 5.
- [Barbará et. al, 2003] Barbará, S., Valle, R. e Mahler, C. (2003). A Contribuição do CMMI na Gestão dos Processos da Indústria de Software do Brasil, China e Índia: experiências e práticas. IEEE Software.
- [Bas Vodde, 2005] Bas Vodde (2005). Nokia Networks and Agile Development. EuroMicro Conference 2005.
- [Basri e O'Connor, 2010] Basri, S. e O'Connor, V. (2010). Organizational Commitment Towards Software Process Improvement: An Irish Software VSEs Case Study, Proc. ITSIM'10, IEEE, 2010, pp.1456-1461.
- [Beck, 2004] Beck, K. (2004) Extreme Programming Explained: Embrace Change 2ª Ed. Addison-Wesley.
- [Brereton et al, 2008] Brereton, P., B. Kitchenham, D. Budgen, e Z. Li. (2008). Using a protocol template for case study planning. in Evaluation and Assessment in Software Engineering (EASE'08). 2008. Bari, Italia p. 1-8.
- [Birk & Rombach, 2001] Birk, A. & Rombach, H. (2001). A Practical Approach to Continuous Improvement in Software Engineering. Software Quality. State of the Art in Management, Testing, and Tools, Berlin Springer.
- [Boehm, 2002] Boehm, B. (2002). Get ready for agile methods, with care.

Computer 55(1): pp 64-69.

**[Boehm & Turner, 2003]** Boehm, B. & Turner, R. (2003). Balancing Agile and Discipline: A guide for the perplexed. Addison-Wesley.

**[Boehm & Turner, 2010]** Boehm, B. & Turner, R. (2010). Love and Marriage: CMMI and Agile Need Each Other. 2010.

**[Bos & Vriens, 2004]** Bos, E., Vriens, C. (2004). An agile CMM. In proceedings of the Extreme Programming and Agile Process in Software Engineering.

**[Briand et al., 1995]** Briand, L., El Aman, K., Melo, W. (1995). ANSI An Inductive Method for Software Process Improvement: Concrete Steps and Guidelines. In Proceedings of the ESI-ISCN95: Measurement and Training Based Process Improvement, Sep. 11-12 1995, Vienna, Austria.

**[Butt 2007]** Butt, A. J. (2007). Identifying Factors Affecting Software Process Improvement during Change. Master Thesis. Disponível em:

[http://www.bth.se/fou/cuppsats.nsf/all/a9085ec1c422bfeec125740a005ae861/\\$file/Thesis.pdf](http://www.bth.se/fou/cuppsats.nsf/all/a9085ec1c422bfeec125740a005ae861/$file/Thesis.pdf) last access in 10/05/2012.

**[Catunda et al., 2010]** Catunda, E., Nascimento, C., Cerdeira, C.T. (2010). Implementando o nível F do MR-MPS com práticas da metodologia ágil Scrum". In: VI Workshop Anual do MPS, v. 1, pp. 78-87, Campinas, São Paulo.

**[Catunda et al., 2010]** Catunda, E., Nascimento, C., Cerdeira, C.T. (2011). Implementação do Nível F do MR-MPS com Práticas Ágeis do Scrum em uma Fábrica de Software". In: X Simpósio Brasileiro de Qualidade de Software (SBQSo11), Curitiba ó Brasil.

**[Chrissis et al., 2011]** Chrissis, M. B., Konrad, M. & Shrum, S. (2011). CMMI for Development®: Guidelines for Process Integration and Product Improvement, 3rd Edition, Published Mar 10, 2011 by Addison-Wesley Professional. Part of the SEI Series in Software Engineering series.

**[Cohen et al., 2004]** Cohen, D., Lindvall, M. & Costa, P. (2004). An Introduction to Agile Methods. Elsevier Academic Press. 0-12-012162 2-67.

**[Coleman, 2006]** Coleman, G. (2006). Investigating Software Process in Practice: A Grounded Theory Perspective, PhD Thesis DCU, DCU.

**[Coleman e Connor, 2007]** Coleman G. e O'Connor R. (2007). Using grounded theory to understand software process improvement: A case study of Irish software product companies, Information and Software Technology, Vol. 49, No. 6, 531-694.

**[Coleman e Connor, 2008]** Coleman, G. e O'Connor, R. (2008) Investigating Software Process in Practice: A Grounded Theory Perspective, Journal of Systems and Software, Vol. 81, No. 5; pp 772-784.

**[Colla e Montagna, 2008]** Colla, P. E. e Montagna, J. M. (2008). Framework to Evaluate Software Process Improvement in Small Organizations, International Conference on Software Process, Germany.

**[Crosby, 1979]** Crosby, P. (1979). Quality is Free. New York: McGraw-Hill;

**[Dan Turk & Rumpe]** Dan Turk, R. F. & Rumpe, B. (2002). Limitations of Agile Software Processes. In 3<sup>rd</sup> International Conference on Extreme Programming and Agile Process in Software Engineering (XP 2002). Disponível <http://www4.in.tum.de/publ/papers/XP02.Limitations.pdf>

**[Dingsøyr et al, 2012]** Dingsøyr, T. Nerur, P. Balijepallyd, P., Moe, B. (2012) A decade of agile methodologies: Towards explaining agile software development. Journal of Systems and Software Volume 85, Issue 6, June 2012, Pages 12136

1221.

**[Dyba, 2000]** Dyba, T. (2000) Improvisation in small software organizations. *IEEE Software* 17(5): 82-87.

**[Dyba et al., 2005]** Dyba, T., Kitchenham, B. & Jørgensen, M. (2005). Evidence-based software engineering for practitioners, *IEEE Software* 22 (1); pp 58665.

**[Dyba e Dingsoyr, 2008]** Dybå, T., and Dingsøy, T. (2008). Empirical Studies of Agile Software Development: A Systematic Review, *Information and Software Technology* (50:9-10), pp. 833-859.

**[Dorling, 2010]** Dorling, A. (2010). Adaptive Process Improvement. Keynotes of Software Process Improvement and Capability Determination conference (Spice 2010).

**[Erickson et al., 2005]** Erickson, J., Lyytinen, K. & Siau, K. (2005). Agile modeling, agile software development and extreme programming. The state of research. *Journal of Database Management* 16(4): pp 115-118.

**[Evered e Lous, 1981]** Evered, R. e Lous, M. R. (1981). Alternative perspectives in the organizational sciences: "Inquiry from the inside" and "inquiry from the outside", *Academy of Management Review* 6(3): 385-395.

**[Flyvbjerg, 2001]** Flyvbjerg, B. (2001). Making social science matter: Why social inquiry fails and how it can succeed again. Cambridge, UK: Cambridge University Press.

**[Fowler, 2002]** Fowler, M. (2002). The agile manifesto: Where it came from and where it may go. Disponível <http://martinfowler.com/articles/agileStory.html>. Último Acesso em 21/04/2012.

**[Fritzsche & Keil, 2007]** Fritzsche, M., Keil, P. (2007) Agile Methods and CMMI: Compatibility or Conflict. *e-Informatica Software Engineering Journal* 1(1), 9626.

**[Fuggetta, 2000]** Fuggetta, A. (2000). Software process: a roadmap. In *Proceedings Conference on The Future of Software Engineering*, Limerick, Ireland; pp. 25634.

**[Fægri, 2012]** Fægri, T. E. (2012). Trends within Software Process Improvement. PHD Thesis.

**[Gergiadou et al., 2003]** Georgiadou, E., Siakas, K., & Berki, E. (2003). Quality improvement through the identification of controllable and uncontrollable factors in software development. In *EuroSPI 2003 (European Software Process Improvement Conference)* (pp. IX-45), Graz, Austria, 10-12.

**[Gallagher et al, 2011].** Gallagher, B., Phillips, M., Richter, K. & Shrum, S. (2011) *CMMI for Acquisition: Guidelines for Improving the Acquisition of Products and Services* 2nd Edition. Addison Wesley.

**[Glazer, 2001]** Glazer, H. (2001). Dispelling the Process Myth: Having a Process Does Not Mean Sacrificing Agility or Creativity. *CrossTalk. The Journal of Defense Software Engineering*: pp. 27--30.

**[Glazer et al., 2008]** Glazer, H., Dalton, J., Andreson, D., Konrad, M., Shrum, S. (2008). CMMI or Agile: Why Not Embrace Both! Technical Note CMU/SEI 2008-TN-003. Carnegie Mellon University - Software Engineering Institute.

**[Gorschek e Wohlin, 2004]** Gorschek, T. e Wohlin, C. (2004) Packaging software process improvement issues: a method and a case study. *Software: Practice and Experience*. vol. 34, no. 14. pp. 131161344.

**[Goulding, 2002]** Goulding, C. (2002). *Grounded Theory: A Practical Guide for Management, Business and Market Researchers*, Sage Publications.

- [Gupta, 2004] Gupta, P. (2004). The Six Sigma Performance Handbook: A Statistical Guide to Optimizing Results. McGraw-Hill Professional. 512p.
- [Herbsleb et al., 1997] Herbsleb, J., Zubrow, D., Goldenson, D., Hayes, W. e Paulk, M. (1997). Software quality and the capability maturity model. Communications of the ACM 40(6): 30640.
- [Highsmith, 1999] Highsmith, J.A. (1999). Adaptive Management: Patterns for the E-Business era, Cutter IT Journal, September.
- [Highsmith, 2002a] Highsmith, J. (2002). Agile software development: Why it is hot! pp. 1-22.
- [Highsmith 2002b] Highsmith, J. (2002). Agile Software Development Ecosystems. The Agile Software Development Series, ed. A. Cockburn and J. Highsmith. Boston, MA: Addison-Wesley.
- [Highsmith, 2011] Highsmith, J. (2011). Stop Doing Agile, Start Being Agile. Webminar Disponível em: <http://www.thoughtworks.com/continuous-delivery/events/stop-doing-agile-start-being-agile>
- [Highsmith e Cockburn, 2001] Highsmith, J. e Cockburn, A. (2001). "Agile Software Development: The Business of Innovation," Computer, vol. 34, pp. 120-122.
- [Humphrey, 1989] Humphrey, W.S. (1989) Managing Software Process, Addison Wesley, Reading, MA.
- [Hurtado & Bastarrica, 2006] Hurtado, A. & Bastarrica, M. (2006). Implementing CMMI using a Combination of Agile Methods, CLEI.
- [Iacocca, 1991] Iacocca Institute. (1991). 21st Century Manufacturing Enterprise Strategy, Lehigh University, Bethlehem, PA.
- [IBGE, 2009] Instituto Brasileiro de Geografia e Estatística (2009) Censo Agropecuário - Agricultura Familiar 2006. Agricultura familiar ocupava 84,4% dos estabelecimentos agropecuários. IBGE. Comunicação Social, 30 de setembro de 2009. Disponível em: <http://www.ibge.gov.br/home/presidencia/noticias/noticiaimpressao.php?idnoticia=1466> Último Acesso em 12/07/2012
- [ISO, 1995] ISO/IEC 12207. Information technology ó software life-cycle processes, 1995.
- [Jakobsen & Johnson, 2008] Jakobsen, C., Johnson. K., (2008). Mature Agile with a twist of CMMI. In *proceedings of the Agile Development Conference*; pp. 212-217.
- [Jakobsen & Sutherland, 2009] Jakobsen, C. & Sutherland, J. (2009). Scrum and CMMI ó Going from Good to Great: are you ready-ready to be done-done?," in Agile 2009, Chicago.
- [Kähkönen & Abrahamsson, 2004] Kähkönen, T., Abrahamsson. P. (2004). Achieving CMMI Level 2 with Enhanced Extreme Programming Approach. In *Lecture Notes in Computer Science of Product Focused Software Process Improvement*; pp 378-392.
- [Keenan, 2004] Keenan, F. (2004): Agile process tailoring and problem analysis (apty). In: *Proceedings of the 26th International Conference on Software Engineering (ICSE04)*.
- [Kerth, 2001] Kerth, N. L. (2001). Project Retrospectives: A Handbook for Team Reviews. Dorset House Publishing.
- [Kitchenham & Pfleeger, 1996] Kitchenham, B., e Pfleeger, S. L. (1996). Software quality: The elusive target. IEEE Software, 13(1), 12-21.

- [**Kitchennam & Pfleeger, 2002a**] Pfleeger, S., e Kitchenham, B. (2002). Principles of survey research Part 2: designing a survey. Software Engineering Notes 27(1).
- [**Kitchenham & Pfleeger, 2002b**] Pfleeger, S., e Kitchenham, B. (2002). Principles of survey research Part 3: constructing a survey instrument. Software Engineering Notes 27(2).
- [**Kitchenham & Pfleeger, 2002c**] Pfleeger, S., e Kitchenham, B. (2002). Principles of survey research Part 4: questionnaire evaluation. Software Engineering Notes 27(3).
- [**Kitchenham & Pfleeger, 2002d**] Pfleeger, S., e Kitchenham, B. (2002). Principles of survey research Part 5: populations and samples. Software Engineering Notes 27(5).
- [**Kitchenham, Pickard e Pfleeger, 1995**] Kitchenham, B., Pickard, L. e Pfleeger, S. L. (1995). Case studies for method and tools evaluation. IEEE Software, 12(4), 52-62.
- [**Klein e Myers, 1999**] Klein, H. e Myers, D. (1999). A set of principles for conducting and evaluating interpretive field studies in information systems, MIS Quarterly 23(1): 67-94.
- [**Koch, 2005**] Koch, A. S. (2005). Agile Software Development - Evaluating the Methods for Your Organization. Artech House, Boston.
- [**Konrad & Over, 2005**] Konrad, M. & Over, J. W. (2005). Agile CMMI: no oxymoron. Software development magazine, February.
- [**Kovacheva & Todorov, 2011**] Kovacheva, T. & Todorov, N. (2011) Optimizing software development process: A case study for integrated Agile-CMMI process model. In International Conference on Computer as a Tool.
- [**Krasner 1999**] Krasner, H. (1999). The Payoff for Software Process Improvement: What It Is and How to Get It. In K. El Emam and N. H. Madhavji (eds.): Elements of Software Process Assessment and Improvement, Los Alamitos, CA: IEEE Computer Society Press.
- [**Kruchten, 2011**] Kruchten, P. (2011). Agile's Teenage Crisis?. Disponível em: <http://www.infoq.com/articles/agile-teenage-crisis>
- [**Langley, 1999**] Langley, A. (1999) Strategies for theorizing from process data. Academy of Management Review, 24; pp-691-710.
- [**Lycett et al., 2003**] Lycett, M., Macredie, R. D., Patel, C. & Paul, R. J. (2003). Migrating agile methods to standardized development practice. Computer 36(6), pp 79-85.
- [**McMahon, 2010**] McMahon, P. (2010). Integrating CMMI® and Agile Development, Addison-Wesley, ISBN: 978-0-321-71410-7.
- [**Marçal et al., 2007**] Marçal, A., Freitas, B., Furtado, F., Belchior, A., (2007). Mapping CMMI Project Management Process Areas to SCRUM Practices. In proceedings of the 31st IEEE Software Engineering Workshop; pp. 13-22.
- [**Marçal et al., 2008**] Marçal, A., Freitas, B., Furtado, F., Maciel, T., Belchior, T. (2008) Blending Scrum practices and CMMI project management process areas. In Innovations in Systems and Software Engineering Journal; pp 17-29.
- [**Martin e Turner, 1986**] Martin, P.Y., Turner, B.A., 1986, Grounded Theory and Organizational Research, The Journal of Applied Behavioral Science, Vol. 22. No. 2., 141-157.

- [**Marinsson, 2003**] Marinsson, J. (2003) Maturing XP through the CMM. In proceedings of the Extreme Programming and Agile Process in Software Engineering.
- [**Mathiassen, Ngwenyama & Aen, 2005**] Mathiassen, L., Ngwenyama, O., Ko, & Aen, I. (2005). Managing changes in software process improvement. *IEEE Software* 22(6): pp 84-91.
- [**McMahon, 2010**] McMahon, P. (2010). Integrating CMMI and Agile Development: Case Studies and Proven Techniques for Faster Performance Improvement (SEI Series in Software Engineering). Addison Wesley. 368p.
- [**Mintzberg e McHugh, 1985**] Mintzberg, H. e McHugh, A. (1985). Strategy in an adhocracy. *Administrative Science Quarterly*, 30, 160-197.
- [**Moe et al., 2010**] Moe, N. B., Dingsoyr, T., & Dyba, T. A. (2010) teamwork model for understanding an Agile team: A case study of a Scrum project. *Information and Software Technology* 52(5); pp 480-491.
- [**Moe, 2011**] Moe, N. B. (2011). From Improving Processes to Improving Practice. PHD Thesis, Disponível em: <http://ntnu.diva-portal.org/smash/get/diva2:438637/FULLTEXT02>. Último Acesso em 10/05/2012.
- [**MPS.BR, 2011a**] MPS.BR ó Melhoria de Processo do Software Brasileiro, Guia Geral (Versão 1.0), disponível em: <http://www.softex.br/mpsbr/guias/guias/MPS.BRGuiaGeral2011.pdf>
- [**MPS.BR, 2011b**] Melhoria de Processo do Software Brasileiro, Guia de aquisição 2011 disponível em: <http://www.softex.br/mpsbr/guias/guias/MPS.BRGuiadeAquisicao2011.pdf>
- [**Naur & Randell, 1968**] Naur, P. & Randell, B. (1968). Software engineering: Report on a Conference Sponsored by the NATO Science Committee. Paper presented at the 1st NATO Software Engineering Conference, Garmisch, Germany. October.
- [**Nawrocky et al., 2002**] Nawrocky, J., Walter, B., Wojciechowski, B. (2002) Comparison of CMM Level 2 and eXtreme Programming. In Lecture Notes in Computer Science of Software Quality.
- [**Nerur et al., 2005**] Nerur, S., Mahapatra, S. & Mangalaraj, G. (2005). Challenges of migrating to agile methodologies. In *Communication of the ACM* 48(5): pp 72-78.
- [**Nerur & Balijepally, 2007**] Nerur, S. & Balijepally, V. (2007). Theoretical reflections on agile software methodologies ó The traditional goal of optimization and control is making way for learning and innovation. In *Communication of the ACM* 50(3): pp 79-83.
- [**Oliveira et al., 2007**] Oliveira, A. C. G., Guimarães, F. A., Fonseca, I. A. (2007). Utilizando Metodologias Ágeis para atingir MPS.BR nível F na Powerlogic. *Revista Visão Ágil*, 3.ed.
- [**Oktaba et al., 2006**] Oktaba, H., García, F., Piattini, P., Ruiz, F., Pino, F. & Alquicira, C. . )2006). MoProSoft1: A Software Process Model for Small Enterprises. Proceedings of the First International Research Workshop for Process Improvement in Small Settings. January; pp. 93-101
- [**Oza e Abrahamsom, 2009**] Oza, N. e Abrahamsom, P. (2009). Building Blocks of Agile Innovation. BookSurge Publishing. 220p.
- [**Paulk, 2001**] Paulk. M. C. (2001). Extreme Programming from a CMM perspective, *IEEE*, vol. 18.

- [**Perry et al., 2000**] Perry, E., Porter, A., e Votta, L.. (2000). Empirical studies of software engineering: a roadmap. In Proceedings of the Conference on The Future of Software Engineering (ICSE '00). ACM, New York, NY, USA, 345-355. DOI=10.1145/336512.336586.
- [**Pettersson et al., 2008**] Pettersson, F., Ivarsson, M., Gorschek, T., Öhman, P. (2008) A practitioner's guide to light weight software process assessment and improvement planning. The Journal of Systems and Software, vol. 81, no. 6; pp. 9726995.
- [**Pfleeger e Kitchenham, 2001**] Pfleeger, S., e Kitchenham, B. (2001). Principles of survey research Part 1: Turning lemons into lemonade. Software Engineering Notes 26(6) 16Y18.
- [**Pikkarainen, 2008**] Pikkarainen, M. (2008). Towards a Framework for Improving Software Development Process Mediated with CMMI Goals and Agile Practices Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies. In proceedings of the Extreme Programming and Agile Process in Software Engineering; pp-213-214.
- [**Pikkarainen & Mäntyniemi, 2006**] Pikkarainen, M., Mäntyniemi, A. (2006). Approach for Using CMMI in Agile Software Development Assessments: Experiences from Three Case Studies. In proceedings of the Extreme Programming and Agile Process in Software Engineering; pp-213-214.
- [**Pino et al., 2010**] Pino, F. J., Pedreira, O., Garcia, F., Luaces, M. R. & Piattini, M. (2010). Using Scrum to guide the execution of software process improvement in small organizations, Journal of Systems and Software, Vol. 83(10); pp. 1662-1677.
- [**PMI, 2003**] PMI. (2003). Organizational Project Management Maturity Model (Opm3): Knowledge Foundation. Project Management Institute.
- [**Poppendieck e Poppendieck, 2003**] Poppendieck, M. e Poppendieck, T. (2003) Lean Software Development ó An Agile Toolkit for Software Development Managers, Addison-Wesley, Boston, ISBN 0-321-15078-3.
- [**Porrts, 1993**] Potts, C. (1993). Software Engineering Research Revisited. IEEE Software Vol. 10 No 5, September 1993, pp 19-28.
- [**Pries-Heje & Johassen, 2010**] J. Pries-Heje & J. Johansen. (2010). SPI Manifesto, eurospi.net, version A.
- [**Pyzdek, 1992**] Pyzdek, T. (1992). To improve your process ó Keep it simple. IEEE software 9(5): pp-112-113.
- [**Qumer & Henderson-Sellers, 2008**] Qumer, A. & Henderson-Sellers, B. (2008). A framework to support evaluation, adoption and improvement in agile methods in practice. Journal of Systems and Software 81(11); pp-1899-1919.
- [**Rainer e Hall, 2002**] Rainer, A., e Hall, T., (2002). Key success factors for implementing software process improvement: a maturity-based analysis. Journal of Systems and Software (62), 71684.
- [**Redwine et al., 1984**] Redwine, S.T., Becker, L., Marmor-Squires, A. Martin, J. Nash, S.e Riddle. W. (1984) DoD Related Software Technology Requirements, Practices, and Prospects for the Future. IDA Paper P-1788, Institute for Defense Analyses, Alexandria, Virginia, June.
- [**Redwine e Riddle, 1985**] Redwine, S. e Riddle. W. (1985) Software

technology maturation. Proceedings of the Eighth International Conference on Software Engineering, May, pp. 189-200.

[Reifer, 2003] Reifer, D. (2003) XP and CMM. In Innovative Technology for Computer Professional Magazine; pp 14-15.

[Rising & Janoff, 2000] Rising, L., & Janoff, N. (2000) The Scrum Software Development Process for Small Teams. IEEE Software,

[Rocha et al., 2005] Rocha, A., Montoni, M., Santos, G., Oliveira, K., Nathaly, A., Mian, P., Conte, T., Mafra, S., Barreto, A., Albuquerque, A., Figueiredo, S., Soares, A., Bianchi, S., Cabral, R. e Dias, A. (2005) Fatores de Sucesso e Dificuldades na Implementação de Processos de Software Utilizando o MR-MPS e o CMMI, In: I Workshop de Implementadores MPS.BR, Brasília, Brasil. [Rocha et al., 2009] Rocha, A. R., Rubinstein, A., Magalhães, A. L., Katsurayama, A. E., Duque, A., Palestino, C. B., Souza, C., Cerdeiral, C., Teixeira, L., Paiva, N. S. e Barros, L. (2009). Avaliação Conjunta CMMI Nível 3 e MPS Nível C: Lições Aprendidas e Recomendações. V workshop Anual do MPS (WAMPS 2009).

[Ronkko et al., 2011] Rönkkö, M. Peltonen, J. & Frühwirth, C. (2011). Examining the Effects of Agile Methods and Process Maturity on Software Product Development Performance. Software Bussines.

[Salgado et al, 2010] Salgado, A., Melcop, T., Acchar, J. , Rego, P.A., Ferreira, A.I.F, Katsurayama, A.E., Montoni, e M., Zanetti, D. (2010). Aplicação de um Processo Ágil para Implantação de Processos de Software baseado em Scrum na Chemtech, In: IX Simpósio Brasileiro de Qualidade de Software, Belém, Brasil. pp. 351-358.

[Salo, 2007] Salo, O. (2007). Enabling Software Process Improvement in Agile Software Development Teams and Organisations. PHD Thesis. Disponível: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.96.7837&rep=rep1&type=pdf>. Último Acesso em 12/05/2012.

[Salo & Abrahamsson, 2005] Salo, O & Abrahamsson, P. (2005). Integrating Agile Software Development and Software Process Improvement: a Longitudinal Case Study. IEEE computers. 2005; pp. 193-202.

[Santana et al, 2006] Santana, C., Timoteo, A. & Vasconcelos, A. (2006). Mapeamento do modelo de Melhoria do Processo de Software Brasileiro (MPS.Br) para empresas que utilizam Extreme Programming (XP) como metodologia de desenvolvimento. In Simpósio Brasileiro de Qualidade de Software (SBQS 2006).

[Santana et al, 2007] Santana, C., Gusmão, C., Vasconcelos, A. & Timoteo, A (2007). Implantação de um Processo de Estimativas de Tempo em uma Instituição Pública baseado em Pontos por Estória. In Simpósio Brasileiro de Qualidade de Software (SBQS 2007).

[Santana et al., 2008] Santana, C., Gusmão, C., Vasconcelos, A. & Rouiller, A. (2008). Achieving Software Quality Certifications Through Agile Software Development. In international conference on agile manufacture (ICAM 2008).

[Santana et al., 2009a] Santana, C., Gusmão, C., Vasconcelos, A. & Pinheiro, C. (2009). Agile and CMMI what do we know about dancing with elephant. In international conference on agile process an extreme programming (XP 2009).

[Santana et al., 2009b] Santana, C., Gusmão, C., Vasconcelos, A. & Rouiller, A. (2009). Utilização de Práticas Genéricas do CMMI para apoiar a utilização de

metodologias Ágeis. In Conferência Íbero-Americana de Engenharia de Software (IDEAS 2009).

[Santana et al., 2010] Santana, C., Cervino, M., Caetano, D. & Gusmão, C. (2010). Software Process Improvement in Agile Software Development. In 1º Workshop Brasileiro de Metodologias Ágeis (WBMA-2010).

[Santana et al., 2011a] Santana, C., Leoneo, F., Gusmão, C. & Vasconcelos, A. (2011). Using Function Points in Agile Projects. In international Conference on Extreme Programming and Agile Process in Software Engineering (XP 2011).

[Santana et al., 2011b] Santana, C., Caetano, D., Gusmão, C. & Vasconcelos, A. (2011). The Confidence of Agile Assessment Methods in the Context of Software Process Improvement. In Agile 2011 Conference (Agile 2011).

[Santana et al., 2012] Santana, C., Filho, H., Gusmão, C. & Vasconcelos, A. (2012). Does Agile Development Fit in the Actual Context of Software Quality? In Agile India 2012 Conference.

[Salo & Abrahamsson, 2006] Salo, O & Abrahamsson, P. (2006). An Iterative Improvement Process for Agile Software Development. Software Process Improvement and Practice.

[Schreiber, 2001] Schreiber, R.S. (2001). The -How To of Grounded Theory: Avoiding the Pitfalls, in Using Grounded Theory in Nursing, Schreiber, R.S. and Noerager Stern, P. (Eds.), Springer.

[Schwaber e Beedle, 2001] Schwaber, K., e Beedle, M. (2001). Agile Software Development with Scrum. Prentice Hall.

[SEBRAE, 2001] Serviço Brasileiro de Apoio a Pequenas Empresa. (2011). As Pequenas Empresas do Simples Nacional. Disponível em: [http://www.biblioteca.sebrae.com.br/bds/bds.nsf/9BB59A59F0E2E04583257957004777CE/\\$File/NT000470DE.pdf](http://www.biblioteca.sebrae.com.br/bds/bds.nsf/9BB59A59F0E2E04583257957004777CE/$File/NT000470DE.pdf) Último Acesso em 12/07/2012.

[SEI, 2011] Software Engineering Institute. Appraisal Requirements for CMMI® Version 1.3 (ARC, V1.3). Available in: <http://www.sei.cmu.edu/reports/11tr006.pdf>

[SEI, 2012] Software Engineering Institute (2012). Published CMMI® Appraisal Results. Disponível em: <https://sas.sei.cmu.edu/pars/pars.aspx>

[Shaw, 1990] Shaw, M. (1990). Prospects for an Engineering Discipline of Software. IEEE Software, 7(6): 15-24.

[Shaw, 2002] Shaw, M. (2002). What makes good research in software Engineering? International Journal Software Tools and Technology Transfer 4(1):167. ISSN 1433-2779 (print), 1433-2787 (electronic).

[Sheldon, 1998] Sheldon, L., (1998). Grounded theory: issues for research in nursing, Nursing Standard. Vol. 12, No. 52, pp 47-50.

[Shore e Warden, 2008] Shore, J. e Warden, S. (2008). The Art of Agile Development, O'Reilly, 2008, pp. 214.

[Siakas et al., 2005] Siakas Kerstin V., Georgiadou Elli and Berki Eleni (2005). Agile Methodologies and Software Process Improvement. Isaias, P., Baptista-Nunes, M. & Palma, dos Reis A. (Eds), Proceedings of the Virtual Multi Conference on Computer Science and Information Systems (MCCSIS 2005) - Software Engineering and Applications (SEA). Apr 11-28. International Association for the Development of Information Society (IADIS). Portugal. pp.-412-417.

[Sidky, 2007] Sidky, A. (2007). Assessing Readiness for Agile Adoption using a Practical and Innovative Approach. Agile 2007 Conference.

[Smite, Moe e Agerfark, 2010] White, D., Moe, B. e Ågerfalk, P. J. (2010).

- Agility Across Time and Space: Implementing Agile Methods in Global Software Projects. Springer: New York, London.
- [SOFTEX, 2009] Sociedade Softex. (2009) Software e serviços de ti: A indústria brasileira em perspectiva. Observatório SOFTEX.
- [SOFTEX, 2012] Sociedade Softex. (2012) Avaliações MPS publicadas. Disponível em:  
<http://www.softex.br/mpsbr/avaliacoes/avaliacoesmpsbtotal.pdf>
- [Stacey, 2003] Stacey, R. D. (2003). Strategic Management and Organisational Dynamics: The Challenge of Complexity. Fourth Edition. Financial Times, Prentice Hall.
- [Standish Group, 2012] The Standish Group. (2012). Chaos Manifesto. Disponível em: <https://secure.standishgroup.com/reports/reports.php>
- [Staples e Niazi, 2007] Staples, M., e Niazi, M. (2007). Systematic review of organizational motivations for adopting CMM-based SPI. Information and Software Technology, doi:10.1016/j.infsof.2007.07.003.
- [Staples et al., 2007] Staples, M., Niazi, M., Jeffery, R., Abrahams, A., Byatt, P. & Murphy, R. (2007). An exploratory study of why organizations do not adopt CMMI. The Journal of Systems and Software 80 (6); pp - 8836895.
- [Strauss & Corbin, 1998] Strauss, A. & Corbin, J. (1998). Basics of qualitative research: Grounded theory procedures and techniques. Sage Publications.
- [Sutherland et al., 2007] Sutherland, J. Jakobsen, C., Johnson. K. (2007). Scrum and CMMI Level 5: The Magic Potion for Code Warriors. In proceedings of the Agile Development Conference; pp. 466-471.
- [Sutherland, 2009] J. Sutherland, Nokia Test: aka the scrum butt test, <http://jeffsutherland.com/nokiatest.pdf>. last access in January 2012.
- [Tarawneh et al., 2011] Tarawneh, M., Abdullah, M. & Ali, A. (2011). A proposed methodology forestablishing software process development improvement for small software development firms. Proceedings of Computer Science, vol. 3, pp. 893 ó 897, 2011, world Conference on InformationTechnology. Available:  
<http://www.sciencedirect.com/science/article/B9865-527GFKD-12/2/913f8b1a5fdb02a7106b94ed3853afd6>
- [Theunissen et al., 2003] Theunissen, W.H.M., Kourie, D.G. & Watson, B.W. (2003) Standards and Agile Software Development. In: Proceedings of SAICSIT 2003; pp. 1611.
- [Travassos & Kalinowsky, 2008] Travassos, G. & Kalinowsky, M. (2008). Variação de desempenho nas empresas que adotaram o modelo mps: resultados. Disponível em: <http://www.softex.br/mpsbr/livros/imps/imps.pdf>
- [Travassos & Kalinowsky, 2009] Travassos, G. & Kalinowsky, M. (2009). Variação de desempenho nas empresas que adotaram o modelo mps: resultados iniciais imps 2009. No 5º WORKSHOP ANUAL DO MPS (WAMPS 2009).
- [Travassos, 2010] Travassos, G. (2010). Resultados Iniciais iMPS 2010. Variação de desempenho nas empresas que adotaram o modelo mps: no 6º WORKSHOP ANUAL DO MPS (WAMPS 2010).
- [Travassos, 2011] Travassos, G. (2011). iMPS ó Rodada 4: Variação de Desempenho nas Empresas que Adotaram o Modelo MPS de 2008 a 2011ö: no 7º WORKSHOP ANUAL DO MPS (WAMPS 2011).

- [**Trujano et al., 2010**] Trujano, M. M., Oktaba, H., Pino, F. & Orozco, M. (2010). Applying Agile and Lean Practices in a Software Development Project into a CMMI Organization. In PROFES 2010.
- [**Turk et al., 2002**] Turk D., France R., Rumpe B. (2002). Limitations of agile software processes, in: Proc.3rd International Conference on eXtreme Programming and Agile Processes in Software Engineeringô XP2002. Disponível em:  
<http://www4.informatik.tu-muenchen.de/~rumpe/ps/XP02.Limitations.pdf>
- [**Turner, 2002**] Turner, R. (2002). Agile Development: Good Process or Bad Attitude? In Lecture Notes in Computer Science of Product Focused Software Process Improvement; pp. 134-144.
- [**Turner & Jain, 2002**] Turner, R. & Jain, A. (2002). Agile Meets CMMI: Culture Clash or Common Cause. In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods XP/ Agile Universe; pp. 153-165.
- [**Van Loon, 2007**] VAN LOON, H. Process Assessment and ISO/IEC 15504: a reference book. Springer-Verlag, New York: 2007.
- [**Veloso et al., 2003**] Veloso, F., Botelho, A. J. J., Tschang, T., e Amsden, A. (2003). Slicing the knowledge-based economy in Brazil, China and India: a tale of 3 software industries. Report. Massachusetts Institute of Technology (MIT).
- [**VersionOne, 2011**] VersionOne. (2011). State of Agile Survey, sixty annual survey. <http://www.versionone.com/stateofagiledevelopmentsurvey/11/>
- [**Vidgen e Wang, 2006**] Vidgen, R. e Wang, X. (2006): Organizing for Agility: a complex adaptive systems perspective on agile software development process. Proceedings of the Fourteenth European Conference on Information Systems (Ljunberg J., Andersson, M. eds.), 1316-1327, Goteborg, Sweden.
- [**Vinekar et al., 2006**] Vinekar, V., Slinkman, C. & Nerur, S. (2006). Can agile and traditional systems approaches coexists? An ambidextrous view. Information systems management 23(3): pp31-42.
- [**Vriens, 2003**] Vriens. C. (2003). Certifying for CMM Level 2 and ISO9001 with XP@Scrum. In proceedings of the Agile Development Conference; pp. 120-124.
- [**Vriens & Barto, 2008**] Vriens, C. & Barto, R. (2008). 7 Years of Agile Management In proceedings of the Agile Development Conference, pp. 390-394.
- [**Walsham, 1995**] Walsham, G. (1995). The Emergence of Interpretivism in IS Research. Information Systems Research 6(4): pp- 376-394.
- [**Wamps, 2012**] Workshop Annual do MPS. Disponível em: <http://www.softex.br/mpsbr/wamps/default.asp>
- [**Wang e Conboy, 2009**] Wang, X. e K. Conboy. (2009). Understanding Agility In Software Development Through A Complex Adaptive Systems Perspective, ECIS 2009.
- [**Wang e Conboy, 2011**] Wang, X. e K. Conboy. (2011). Comparing Apples with Oranges? The Perceived Differences between Agile and Lean Software Development Processes. ICIS 2011.
- [**Waterman, 1990**] Waterman, R. H. (1990). Adhocracy: The Power to

Change. Knoxville, TN: Whittle Direct.

**[Weber et al., 2000]** Weber, K., Nascimento, C., Marinho, D., Durski, G. e Amaral, H. (2000). Measurements of Quality and Systemic Productivity in the Brazilian Software Industry (Brasilia: Ministerio daCiencia e Tecnologia).

**[Weber et al., 2005]** Weber, C. K., Machado, C. A. F., Scalet, D., Salviano, C. F., & Rocha, A. R. (2005). Modelo de Referência e Método de Avaliação para melhoria do processo de software - versão 1.0 (MR mps e MA mps), IV Simpósio Brasileiro de Qualidade de Software, Porto Alegre ó Brasil.

**[Weber et al., 2008]** Weber, K. C.; Montoni, M.; Rocha, A.R.C.; Santos, G.; Barbieri, C., José Antonio Antonioni, J.A. (2008). õMPS.BR ó Melhoria de Processo do Software Brasileiro: resultados alcançados e lições aprendidas (2004-2008)ö. CLEI 2008 (XXXIV Conf. Latinoamericana de Informática), 8 al 12 de Septiembre, Santa Fé, Argentina.

**[Weber e Amaral, 1999]** Weber, K. e Amaral, H. (1999). õSoftware quality and productivityin Brazilö. Paper presented at the International Software EngineeringStandards Symposium, Curitiba, Brazil, May.164 Transnational Corporations, Vol. 16, No. 1. 2007.

**[Weber e Pinheiro, 1995]** Weber, K. C. e Pinheiro, M. (1995) Software Quality in Brazil. Quality World Magazine, London, v. 21, n.1.

**[Williams, 2012]** Williams, L. (2012). What Agile Teams Think of Agile Principles, Communications of the ACM, Volume 55 Issue 4, April 2012. pp 71-76.

**[Yin, 2002]** Yin, R. K. (2002). Case Study research design and methods. Thousand Oaks, Calif Sage.

## **Apêndice A o Email de Apresentação.**

Prezado.

Meu nome é Célio Santana e estou conduzindo uma pesquisa sobre Melhoria de Processo de Software Ágil em ambientes CMMI. Esta pesquisa visa identificar o perfil desta melhoria (Ágil) dentro de organizações que estão pretendendo se certificar CMMI nos próximos anos.

A intenção dessa pesquisa é avaliar as características desse tipo de Melhoria de Processo e as vantagens e desvantagens da sua utilização. Para tanto será necessário que algumas informações tais como produtividade, qualidade do produto e custo sejam conhecidos pela organização antes do início da implementação do CMMI na empresa e um tempo após a certificação.

A pesquisa será dividida em 3 partes que são (1) levantamento do desempenho atual da empresa, (2) levantamento do desempenho da empresa logo após a certificação, (3) levantamento do desempenho da empresa após 6 meses da certificação (Isso se repetirá a cada seis meses).

Para resguardar as informações da organização, este levantamento não deve ser repassado para mim contendo dados brutos, este levantamento será realizado dentro da organização e as respostas serão de caráter Sim e Não, como por exemplo: A produtividade da sua empresa aumentou em relação a ultima pesquisa? (S/N), contudo se faz necessário que a organização guarde estas informações para posteriores comparações.

O perfil de empresa procurado é aquela que será avaliada CMMI em algum momento futuro, possui as informações de seu desempenho antes de iniciar o programa de MPS e que iniciou a implementação do CMMI utilizando abordagens ágeis.

Para prosseguir com a pesquisa gostaria que fossem respondidas as 3 questões a seguir.

Você gostaria de colaborar com a Pesquisa? (S/N) ó ( )

Data de inicio da implementação do CMMI? (MM/AA) - /



*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

início da utilização de métodos ágeis na empresa? (MM/AA) - /

Qual método ágil está sendo adotado?

Gostaria de Receber o resultados da pesquisa (S/N) ó ( )

Agradeço desde já colaboração

## Apêndice B o 1º Levantamento de Informações

Prezado.

Em primeiro lugar obrigado por participar da pesquisa referente à Melhoria de Processo de Software Ágil em ambientes CMMI. Neste primeiro levantamento nós gostaríamos de identificar algumas informações a respeito da empresa antes de iniciar a implementação do CMMI.

É recomendável considerar o último resultado do desempenho da empresa imediatamente antes do início do processo criado para o CMMI entrar em vigor.

No mesmo email em que receberam este questionário, foi enviado também um outro questionário chamado de Nokia Test. Ele serve para avaliar como times Scrum estão seguindo o processo Scrum dentro de uma organização. É ideal que todos os membros envolvidos com o Scrum respondam o Nokia Test, e o resultado final (Média) é a soma de todas as notas dividido pelo número de questões respondidas.

Quaisquer dúvidas que existentes sobre o preenchimento deste questionário podem ser enviadas para o mesmo email que enviou esta mensagem. Segue as perguntas:

1 ó Qual foi média do Nokia Test da empresa antes de iniciar a implementação do CMMI: ()

2 ó O Gap Analysis do seu processo já foi realizado? A organização possui a informação? (S/N): ()

3 ó A empresa está ciente ou tem alguma ideia de sua produtividade atual, seja em Linhas de Código, Horas, Pontos por Função? (S/N): ()

4 ó A empresa está ciente do tempo médio de duração de seus projetos? (S/N): ()

5 - A empresa está ciente da qualidade do seu produto? Ela é Mensurada de alguma forma (S/N): ()

6 ó Qual foi a Data do início da implementação do MPS? //

## Apêndice C o 2º Levantamento de Informações

Prezado.

Parabéns pelo sucesso na avaliação de sua empresa no CMMI. Sei que este momento é de muito trabalho, entretanto como já foi explicado, eu ainda preciso das informações da sua organização.

Novamente agradeço pela participação na pesquisa. Neste momento é necessário que as informações a respeito do desempenho da empresa imediatamente após o sucesso na certificação CMMI, por isso recomendo que seja usado os resultados da primeira compilação de resultados após o CMMI.

No mesmo email em que receberam este questionário, foi enviado também um outro questionário chamado de Nokia Test. Ele serve para avaliar como times Scrum estão seguindo o processo Scrum dentro de uma organização. É ideal que todos os membros envolvidos com o Scrum respondam o Nokia Test, e o resultado final (Média) é a soma de todas as notas dividido pelo número de questões respondidas.

Quaisquer dúvidas que existentes sobre o preenchimento deste questionário podem ser enviadas para o mesmo email que enviou esta mensagem. Segue as perguntas:

1 ó Qual foi média do Nokia Test da empresa logo após a obtenção do CMMI: ()

2 ó Se compararmos a produtividade atual da empresa com aquela medida antes do início da implementação do CMMI, ela aumentou ou diminuiu? (A/D): ()

3 ó Se compararmos o tempo médio gasto por projeto atualmente na empresa com aquele medido antes do início da implementação do CMMI, ele aumentou ou diminuiu? (A/D): ()

4 ó Se compararmos a qualidade final do produto de software desenvolvido atualmente na empresa com aquele medido antes do início da implementação do CMMI, ele aumentou ou diminuiu? (A/D): ()



*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

5 ó Se compararmos o Faturamento da empresa com aquele medido antes do início da implementação do CMMI, ele aumentou ou diminuiu? (A/D): ( )

6 ó Considerar o faturamento total da empresa desde o início da implementação do CMMI até a data da avaliação, chamaremos esse número de FT. Considere também todos os gastos tidos pela empresa para implementar o CMMI, chamaremos esse de FC). Você é capaz de fornecer o valor numérico da razão FC/FT?

## Apêndice D o 3º Levantamento de Informações

Prezado.

Novamente agradeço pela participação na pesquisa. Neste momento será coletada informações sobre como está o processo desta empresa após 6 meses do sucesso na certificação CMMI. Recomendo que seja usado os resultados mais recentes dos indicadores da organização.

No mesmo email em que receberam este questionário, foi enviado também um outro questionário chamado de Nokia Test. Ele serve para avaliar como times Scrum estão seguindo o processo Scrum dentro de uma organização. É ideal que todos os membros envolvidos com o Scrum respondam o Nokia Test, e o resultado final (Média) é a soma de todas as notas dividido pelo número de questões respondidas.

Quaisquer dúvidas que existentes sobre o preenchimento deste questionário podem ser enviadas para o mesmo email que enviou esta mensagem. Segue as perguntas:

1 ó Qual foi média do Nokia Test da empresa hoje: ()

2 ó Se a empresa fosse ser avaliada hoje no CMMI como ficaria o gráfico de avaliação (Por área de processo)?

3 ó O retorno de Investimento do programa foi maior do que 1?

4 ó O Investimento em melhoria de processo de software hoje está maior do que na época da avaliação CMMI (S/N)? ()

5 ó Se compararmos a produtividade atual da empresa com aquela medida antes do início da implementação do CMMI, ela aumentou ou diminuiu? (A/D): ()

6 ó Se compararmos a produtividade atual da empresa com aquela medida pela ultima vez, ela aumentou ou diminuiu? (A/D): ()

7 ó Se compararmos o tempo médio gasto por projeto atualmente na empresa com aquele medido antes do início da implementação do CMMI, ele

aumentou ou diminuiu? (A/D): ()

8 ó Se compararmos o tempo médio gasto por projeto atualmente na empresa com aquele medido pela última vez, ele aumentou ou diminuiu? (A/D): ()

9 ó Se compararmos a qualidade final do produto de software desenvolvido atualmente na empresa com aquele medido antes do início da implementação do CMMI, ele aumentou ou diminuiu? (A/D): ()

10 ó Se compararmos a qualidade final do produto de software desenvolvido atualmente na empresa com aquele medido pela última vez, ele aumentou ou diminuiu? (A/D): ()

11 ó Se compararmos o custo médio por projeto atualmente na empresa com aquele medido antes do início da implementação do CMMI, ele aumentou ou diminuiu? (A/D): ()

12 ó Se compararmos o custo médio por projeto atualmente na empresa com aquele medido pela última vez, ele aumentou ou diminuiu? (A/D): ()

13 - Como os membros das equipes Scrum percebem a agilidade dentro do processo atual utilizado pela organização?



**PDF**  
Complete

*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

14 - Como os membros das equipes Scrum percebem que eles influenciam na melhoria da organização como um todo?



**PDF**  
Complete

*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)



**PDF**  
Complete

*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

150 Quando a empresa e o time lidam com as não conformidades surgidas no processo?



**PDF**  
Complete

*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

166 Houve algum pedaço do processo criado para o CMMI que foi abandonado pelo time? Por que ele foi abandonado? Como ele foi substituído?



**PDF**  
Complete

*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

## Apêndice E o Empresas que não seguiram com o CMMI.

Prezado.

Novamente agradeço pela participação na pesquisa. Apesar da empresa não ter atingido o objetivo de obter a certificação CMMI é de extrema importância que as impressões obtidas pela sua empresa fiquem registradas. Por isso pedimos que mais duas perguntas sejam respondidas neste momento.

Quaisquer dúvidas que existentes sobre o preenchimento deste questionário podem ser enviadas para o mesmo email que enviou esta mensagem. Segue as perguntas:

1 ó Qual foi a razão que levou a organização a interromper o programa de MPS com foco no CMMI.

- Falta de apoio da alta gestão;
- Mudança de prioridade do programa pela alta gestão;
- Objetivos organizacionais já foram atingidos;
- Falta de visibilidade dos resultados do programa;
- Outro. Qual ();

2óQual o comentário da equipe envolvida sobre a adoção de métodos ágeis nesta implementação do CMMI?



**PDF**  
Complete

*Your complimentary  
use period has ended.  
Thank you for using  
PDF Complete.*

[Click Here to upgrade to  
Unlimited Pages and Expanded Features](#)

## Apêndice F o Artigos Publicados.

Os artigos que foram publicados a partir desta pesquisa são apresentados a seguir conforme a Tabela 13 abaixo:

**Tabela 13 –Artigos Publicados**

Nº	Trabalho	Conferência
P1	Agile Software Development and CMMI: What we do not know about Dancing with Elephants. [Santana et al., 2009a]	XP 2009
P2	Software Process Improvement in Agile Software Development [Santana et al., 2010]	WBMA 2010
P3	Using Function Points in Agile Projects [Santana et al., 2011a]	XP 2011
P4	The Confidence of Agile Assessment Methods in the Context of Software Process Improvement [Santana et al., 2011b]	Agile 2011
P5	Does Agile Development Fit in the Actual Context of Software Quality? [Santana et al., 2012]	Agile India 2012
P6	Utilização de Práticas Genéricas do CMMI para apoiar a utilização de Metodologias Ágeis. [Santana et al, 2009b].	IDEAS 2009

## Agile Software Development and CMMI: What we do not know about Dancing with Elephants

Célio Santana<sup>1</sup>, Cristine Gusmão<sup>1</sup>, Liana Soares<sup>1</sup>, Caryna Pinheiro<sup>2</sup>, Teresa Maciel<sup>3</sup>, Alexandre Vasconcelos<sup>4</sup> and Ana Rouiller<sup>3</sup>

<sup>1</sup>University of Pernambuco, Department of Computer and Systems, Benfica St. 455, 50720-001 Recife, Pernambuco, Brazil

<sup>2</sup>University of Calgary, Department of Computer Science, Calgary, Canada

<sup>3</sup>Federal Rural University of Pernambuco, Informatic and Statistic Department, Dois Irmãos avenue. S/N, 52171-900 Recife, Pernambuco, Brazil

<sup>4</sup>Federal University of Pernambuco, Informatic Centre, Prof Luiz Freire Avenue. S/N, 50740-540 Recife, Pernambuco, Brazil

Celio.santana@gmail.com, capinhei@ucalgary.ca, cmgg@dsc.upe.br, teresa@cesar.org.br, lsoos@dsc.upe.br, [amlv@cin.ufpe.br](mailto:amlv@cin.ufpe.br), anarouiller@gmail.com

**Abstract.** In this paper we discuss how the merging of Agile Methodologies and Software Quality Models in same process today is ignoring many important aspects of both approaches. The inconsideration of these points results in a rigid integration of Agile and Quality Models that limits the full potential of their synergies. Ignoring such important items however does not necessarily means that they are not being utilized in the process, it normally indicates their utilization in an ad-hoc way. To explore this topic, we collected qualitative and quantitative data from literature and two Brazilian companies which work with agile and XP.

**Keywords:** Capability Maturity Model Integration, CMMI, Agile Software Development.

### 1 Introduction

In 2001 the Agile movement proposed a shift in the values of software development process from the mechanistic (i.e., driven by process and rules of science) to the organic (i.e., driven by softer issues of people and their interactions) [1].

Despite the changes in traditional software engineering presented by the Agile methods, in 2001 one of the CMM developers, Mark Paulk, stated that CMM and an Agile method called Extreme Programming (XP) [2] contain ideas that can be synergistic [3].

This paper provides, through the analysis of the quantitative and qualitative data gathered in literature and industry, a further contribution by bringing to attention the relevant disregarded elements when merging Agile and CMMI, as well as presents the consequences of implementing this incomplete merged methodology.

After this introductory section, section 2 explores Agile Methods. In section 3, CMMI is explained; section 4 shows how CMMI and Agile are being merged today; section 5 shows what is fully or partially unknown about this merging and section 6 shows a summary and future works.

## 2 Agile Methodologies

In February 2001, seventeen of the leading developers and proponents of the “Agile methods” proposed a meeting which resulted in four levels of agreement among the participants [4]:

1<sup>st</sup> Level: There was a need for methods designed to respond to change during software projects. Thus, they adopted the term “Agile” to identify such methods. They agreed that the term “Light” was not appropriate because certain projects would not employ a “Light” methodology but could still require agility.

2<sup>nd</sup> Level: The four statements of the “Agile Manifesto” [5]. These four statements capture the core values on which all of the Agile Methods are built, as well as the spirit in which they should be implemented.

3<sup>rd</sup> Level: The next level of agreement was on a set of 12 Agile Principles. In these statements, the values are fleshed out in more detail and given more concrete meaning.

4<sup>th</sup> Level: The final level of agreement, which was to be a more detailed with actual activities or tactics to running projects, was beyond their grasp at the time. They left that fourth level for each agile method to define in its own way. Some of these agile techniques are Extreme Programming (XP) [2] and Scrum [6].

## 3 Capability Maturity Model Integration

The Capability Maturity Model Integration (CMMI) [7] is defined as a process improvement maturity model for the development of products and services. Consisting of 22 process areas where each one area can be summarized to illustrate their components, as shown in Figure 1.

CMMI enables one to approach process improvement and appraisals using two different representations: continuous and staged. The continuous representation enables an organization to select a process area and improve processes related to it using six (0 to 5) capability levels to characterize improvement related to an individual process area. The staged representation uses predefined sets of process areas to define an improvement path for an organization characterized by five (1 to 5) maturity levels.

The following rules summarize equivalent staging:

- To achieve maturity level 2, in seven process areas assigned to maturity level 2 must achieve capability level 2 or higher.

- To achieve maturity level 3 or higher, all process areas assigned to that specific maturity levels and lowers must achieve capability level 3 or higher.

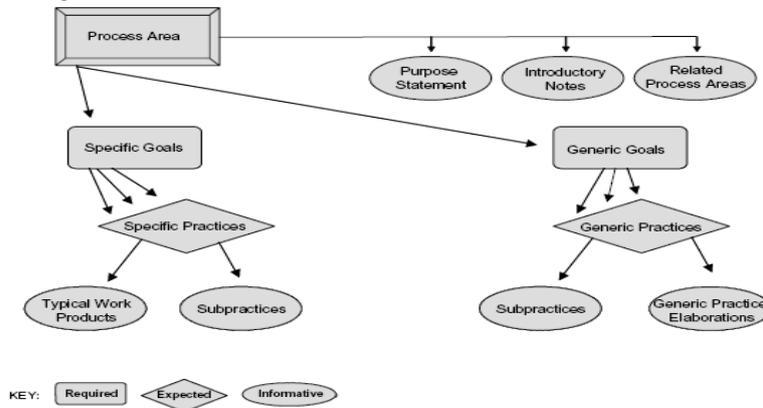


Fig. 1. CMMI Process Areas and their Components [7].

And a short description of each capability level follows:

- A capability level 0 process is defined as an “incomplete process”. One or more of the specific goals of the process area are not satisfied, and no generic goals exist for this level.
- A capability level 1 process is defined as a “performed process.” A performed process is a process that satisfies the specific goals of the process area. Although capability level 1 does not address all generic goals at capability 2.
- A capability level 2 or higher process is a performed (capability level 1) process that address all the generic goals for that capability level.

## 4 Agile and CMMI: What do we Know?

### 4.1 Mapping techniques of Agile Methods to CMMI Specific Practices

The first mapping between Agile and CMM was conducted by Paulk [4] fitting XP to SW-CMM level 2. Since then, many reports following Paulk’s ideas such as [8, 9] were published. These mappings refer to the low levels (2 and 3) of CMMI.

This paper also presents the research conducted in two companies using Agile and following CMMI patterns. Their names must remain anonymous due to confidentiality reasons. Company 1, located in Brazil, has eleven years of experience in the market and has recently been evaluated for SCAMPI to CMMI level 3 and uses Scrum in sixteen projects. Company 2, located in Brazil, is a small-sized company with nine years of experience in the market which will be evaluated at CMMI level 2 using XP in a pilot project. They will be referred in the following sections to provide an industry insight on the subjects.

About this subject, both companies conducted a gap analyses to understand the misfits between CMMI and Agile. These gap analyses are also the mapping of each company about agile and its relationship with CMMI specific practices.

#### **4.2 Traditional and Agile Software Process Improvement**

For Salo & Abrahamsson [10], stated two central differences between traditional and agile SPI. Firstly, the origin of SPI goals is traditionally from the organizational level. However, in the agile project context the power for SPI lies within project teams, and their experiences throughout projects. Secondly, the process knowledge of project teams has traditionally been harvested from finished projects. Instead, the focus of agile project learning is on improving the performance of the ongoing project.

The traditional and agile SPI approaches are not contradictory. Rather, both the approaches can be beneficial in integrating agile software development and organizational learning. This, however, requires the integration of agile SPI mechanisms in the organizational learning cycle.

The companies under study (company 1 and 2) structured their Software Engineering Process Group (SEPG) outside agile teams, following the traditional SPI approaches.

### **5 Agile and CMMI: What we do not Know?**

#### **5.1 CMMI Support for adopting Agile**

According [11, 12], companies using CMMI provide support to agile making it an easier and smoother methodology to adopt than in non CMMI companies. Nevertheless, [9] stated that even presenting similar ideas, merging Agile and CMMI could be difficult, since there are other aspects seen in a different perspective.

This subject looks contradictory because the support provided by CMMI to any methodology comes from its generic practices (see Fig 1). Sutherland et. al. [13] stated how generic practices at capability 2 and 3 could support and improve Agile.

The companies under study (company 1 and 2) ignored how Generic Practices support agile following the industrial tendency of ignore this subject.

#### **5.2 Merging CMMI Specifics Practices and Agile, is still agile at all.**

Mnkandla & Dowlatzky [14] define the value of agility in allowing the concepts defined above to mutate within the parameters set by the agile values and principles. As a matter of fact, there is no mapping that provides a full fit between Agile and CMMI. Thus, it is necessary to complement Agile with other practices, in which there is no guarantee to be following agile values and principles.

Quantitatively, [13] and [15] presented evolution of indicators of quality and productivity with the adoption of Agile merged with CMMI.

Sutherland *et. al.* [13] stated:

1. Scrum reduced defects by almost 50% compared to our previous CMMI level 5 implementation.
2. Projects using Scrum shows a 201% increase in productivity.
3. Price was reduced by 50%.when reevaluate the requirement specification.

Gravis & Jarvistock [15] Stated:

1. Defects reduced by 66%, Critical defects reduced by 79%;
2. Duration (in days) reduced by 44% and Effort (in hours) reduced by 47%;
3. Quality of Life improved by 81% business and 77% technology.

The companies 1 & 2 are still measuring their actual process (Agile + CMMI) to compare against their previous CMMI process.

### 5.3 Aspects almost totally unknown.

- *Adoption of statements and values of Agile Manifesto.*

There is no data in the literature addressing the adoption of values and statements of Agile Manifesto. Neither companies 1 or 2 considered the Manifesto on their processes.

- *Higher maturity levels of CMMI are close or deviate of Agile*

The mappings seen in section 4.1 show to the deviation between agile and process areas and specific practices of CMMI when the maturity level is raised to level 3. However, Sutherland *et. al.* [13] stated that generics practices of level 3 could improve agile. The lack of precise data on how much closer/far are Agile, specifics practices and generics practices makes it impossible to reach a conclusion about this subject.

## 6 Conclusion

Merging Agile e CMMI is neither strange nor innovative, many reports from both academia and industry show how Agile and CMMI could live harmoniously in the same environment with few quantitative results indicating positive results when these approaches work together.

Although, the union between lower levels of Agile Manifesto (agile methods) and CMMI (specific practices), does not provide many options for companies which want to use Agile and CMMI. Ignoring the higher levels of Agile Manifesto and the

others components of CMMI, the companies could not see other opportunities to improve their process, strengthen the union of the approaches and solve commons problems.

Finally there is a few diversity of researches dealing with merging Agile and CMMI. Most of them are mappings where specific practices of the following process areas of CMMI level: Project Planning (PP), Project Monitoring and Control (PMC) and Requirements Management (REQM) are merged with Scrum or XP. This factor leads to a false impression of full understanding about the subject.

### 6.1 Future Works.

- Make further research on how CMMI Generic Practices could support and improve Agile, and establish a “distance” between them;
- Observe the behavior of these distance when higher maturity levels of CMMI are achieved ;
- Collect indicators such as quality, productivity, schedule and costs after the merging and verify their evolution comparing with themselves before the merging;
- Investigate how the Agile Manifesto could be considered in this union.

### References

1. Mnkandla, E., Dwolatzky, B.: Balancing the human and the engineering factors in software development. In: IEEE AFRICON 2004 Conference, pp. 1207–1210 (2004).
2. Beck, K.: Extreme Programming Explained: Embrace Change. Addison Wesley. (2000)
3. Paulk, M.: δXp from a CMM perspective. IEEE Software 18(6), pp 19--26. (2001)
4. Koch. A. S.: Agile Software Development - Evaluating the Methods for Your Organization. Artech House, Boston (2005)
5. Agile Manifesto.: Agile Manifesto for Software Development, <http://www.agilemanifesto.org/> (2001)
6. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall (200)
7. Chrissis, M. B., Konrad, M., Shrum, S.: CMMI@: Guidelines for Process Integration and Product Improvement. Addison Wesley. (2003)
8. Nawrocky, J., Walter, B., Wojciechoeski, A.: Comparison of CMM Level 2 and eXtreme Programming In Lecture Notes in Computer Science of Software Quality. (2002)
9. Kähkönen, T., Abrahamsson, P.: Achieving CMMI Level 2 with Enhanced Extreme Programming Approach. In Lecture Notes in Computer Science of Product Focused Software Process Improvement (PROFES), p. 378–392. (2004)
10. Salo, O., Abrahamsson, P.: Integrating agile software development and software process improvement: a longitudinal case study. International Symposium on Empirical Software Engineering. (2005)
11. Turner, R., Jain, A.: Agile Meets CMMI: Culture Clash or Common Cause. In proceedings of the Second XP Universe and First Agile Universe Conference on Extreme Programming and Agile Methods XP/ Agile Universe. pp. 153--165. (2002)
12. Jakobsen, C., Johnson, K.: Mature Agile with a twist of CMMI. In proceedings of the Agile Development Conference, p. 212--217. (2008)
13. Sutherland, J., Jakobsen, C., Johnson, K.: Scrum and CMMI Level 5: The Magic Potion for Code Warriors. In proceedings of the Agile Development Conference. pp. 466–471. (2007)
14. Mnkandla, E., Dwolatzky, B.: Balancing the human and the engineering factors in software development. In: IEEE AFRICON 2004 Conference, pp. 1207–1210 (2004).
15. Jarvis, B., Gristock, S.: Extreme Programming, Six Sigma & CMMI ó How they can work together, a JP Morgan Chase Study. <http://www.sei.cmu.edu/cmmi/presentations/sep05.presentations/jarvis-gristock.pdf>

## Software Process Improvement in Agile Software Development

Mariana Cerviño<sup>1</sup>, Célio Santana<sup>2,3</sup>, Danilo Caetano<sup>2</sup>, Cristine Gusmão<sup>2</sup>

<sup>1</sup> Departamento de Sistemas Computacionais, Escola Politécnica de Pernambuco (POLI) Universidade de Pernambuco (UPE) Rua Benfica, 445, Madalena ó 50.751-460 Recife, PE ó Brasil

<sup>2</sup> Curso de Sistemas de Informação - Faculdades Integradas Barro Melo Av. Transamazônica, 405, Jardim Brasil ó 53.300-240 Olinda, PE ó Brasil

<sup>3</sup> Centro de Informática ó Universidade Federal de Pernambuco Caixa Postal 7.851 ó 50.732-97 ó Recife ó PE ó Brasil

{maricervino, celio.santana, danilocaetano1987, cristinegusmao} @gmail.com

***Abstract.** Organizations are increasingly adopting agile development methodologies software development along with the traditional. New challenges and opportunities for process improvement software are also emerging. The Nokia Test is an effective method to assess how teams using Scrum are following the procedure proposed by this agile method. Quality and productivity are the key indicators set by the companies that are using the traditional maturity model of process improvement software CMMI. The aim of this study is to analyze the relationship between the traditional methods of improving software process that emphasizes continuous improvement of software processes of organizational development, and principles of the agile approach to software development is iterative, that improvements in software development increasing individual effectiveness in the team.*

**Keywords:** Scrum, Agile software development, Nokia Test, CMMI, Software Process Improvement.

### 1 Introduction

Software process improvement (SPI) has become a practical tool for companies where the quality of the software is of high value [Jarvinem 1994]. In a technical report with results from thirteen organizations, and with the number of post-release defect reports used as a measure, Herbsleb and others practitioners (1994) showed that due to software process improvement (SPI), the products and business value (especially return on investment ó ROI) was improved.

It is generally considered that a well documented and a repeatable process is essential for developing software products of high quality [Stamelos & Sfetsos 2007].

There is also evidence that the use of standards and process assessment models has a positive impact on the quality of the final software product [Kitchenham 1996].

The software engineering community has gradually moved from product quality-centered corrective methods to process quality-centered, preventive methods, thus shifting the emphasis from product quality improvement to process quality improvement. Inspections at the end of the production line have long been replaced by design walkthroughs and built-in quality assurance techniques throughout the development life cycle [Georgiadou 2004].

In traditional SPI methods and approaches the aspect of organizational improvement has usually been placed in a central role, due to the fact that the planning and control of the SPI initiatives are managed by the organizational stakeholders [Basili & Caldiera 1994, SEI 2006]. The reported positive effects of SPI methods and approaches include reducing indicators such as time to market, risks and costs, and increasing the productivity and quality in software development organizations [Krasner 1999, Van Soligen & Berghout 1999].

However, various negative effects have also been encountered, e.g. regarding the cost-effectiveness of SPI initiatives, their actual effectiveness in improving the software development practices of organizations, the volume of the effort needed to implement SPI initiatives and the low speed at which visible and concrete results are achieved [Krasner 1999, Goldensen & Herbsleb 1995, Dybå 2000].

In fact, it has been reported that around two-thirds of SPI traditional initiatives fail to achieve their intended goals [Debou 1999]. From the mid-1990s onwards, agile software development principles and methodologies have been increasingly challenging the traditional view of software development, and therefore, provides a highly untraditional approach to SPI, in which the process improvement knowledge of software developers and software development teams is acknowledged and valued [Salo 2007].

We have long known that skilled people are the most crucial resource in software development. Bach (1994) stated that "Everyone knows the best way to improve software productivity and quality is to focus on people." However, it took more than 10 years for the agile manifesto and agile methods (extreme programming, scrum, crystal, and many others) to truly place the emphasis on people and their interaction [Stamelos & Sfetsos 2007].

This paper presents a state of the art review about SPI in agile software development, this work is part of two MSc dissertations and in this light after this

introductory section, the section 2 explores Scrum, the section 3 explores the Nokia Test, the section 4 shows the relevance of CMMI for this research, in section 5 the goal of the case study and settings of the companies and projects are introduced. Section 6 shows the data retrieved in the companies and their projects, these data will be analyzed using Graphics for better visualization. The Section 7 will be present the conclusions of the paper.

## **2Software Process Improvement**

A software process can be defined as the sequence of steps required to develop or maintain software aiming at providing the technical and management framework for applying methods, tools, and people to the software task [Humphrey 1995]. However, even the most exquisitely defined and managed process may still not meet the context specific needs and objectives of software development organizations and customers regarding, for example, performance, stability, compliance and capability [Florac et al. 2000].

Thus, SPI aims providing software development organizations with mechanisms for evaluating their existing processes, identifying possibilities for improving as well as implementing and evaluating the impact of improvements [Florac et al. 2000].

Traditionally, the ultimate goal of SPI in organizations is to provide a Return on Investment (ROI) for the organization through the improvement activities yielding more money than is spent on them [Rico 2004]. ROI has been reported for various SPI achievements, such as improved efficiency of the development process and reduction of total software costs, increased quality of the end product, higher predictability of cost and schedule, and increased level of reuse (Krasner 1999).

The focus on quality in SPI is based on the fundamental ideology that quality-driven development is likely to yield not only better quality but also lower cost and improvement of competitive position [Deming 1990].

One of the characteristics of SPI, as traditionally defined, is its emphasis on the continuous improvement of organizational software development processes in terms of performance, stability, compliance, and capability, for instance. Often the existing SPI methods and approaches seem to enhance the underlying business goals and needs in the improvement of organizational software development processes [Salo 2007]. Traditionally, SPI initiatives are also strictly controlled and managed by the organizational stakeholders (Boehm & Turner 2003).

Salo (2007) identified six different elements are identified in the context of traditional SPI: (1) The organizational models of continuous improvement, (2) standard processes and assessments, (3) tailoring, (4) deployment, (5) measurement, and (6) the utilization of knowledge and learning in SPI.

The focus of this work is research related to (1) The organizational models of continuous improvement, (2) standard processes and assessments because these subjects were investigated widely in previous researches. The others subjects were not studied or adopted in software industry enough to present a substantial content about the subject, once that researches are just found in Salo (2007) work.

### **3 Software Process Improvement in Agile Context**

Salo (2007) states that agile software development provides new possibilities for conducting SPI, which may well provide grounds for meeting some of the central challenges of traditional SPI. agile software development provides a highly untraditional approach to SPI, in which the process improvement knowledge of software developers and software development teams is acknowledged and valued.

This difference comes from the idea of the traditional goal of a software process is to provide high predictability, stability, and repeatability using highly managed and quantitatively monitored software development processes. On the other hand, agile principles highlight the need for the software process to be flexible, to be able to rapidly respond to the constant changes and context specific needs of software development [Salo 2007].

As a result, traditional software development emphasizes up-front contract negotiations where the requirements, cost and schedule of the product development are fixed and the end product will be delivered at the end of the project lifecycle. In this mode of software development, traditionally, extensive documentation and quantitative monitoring of the product development process plays a central role while. The principles and practices of agile software development, in turn, address the constant changes. [Salo 2007].

#### **3.1 Agile Software Process Improvement Based on Team Behavior (Salo 2007)**

Salo (2007) states when considering the relationship between agile software development and SPI, there are three principles, in particular, of the agile manifesto that deserve attention: the valuing of individuals and interactions over processes and tools, the principle that encourages regular reflection by software development teams in order to become more effective, and the self-organization of software development teams. Taking regular improvement within project teams as one of the twelve principles of agile software development highlights the importance of continuous improvement also in the agile software development context. In order to welcome changes throughout the agile software development project, whether they concern product requirements or technical aspects, the software process with its practices, methods, and tools must be able to adapt to the specific context while also to respond to the changes when needed.

Salo (2007) also states that traditionally, SPI has been approached in a top-down manner, in which the organizational level has played a major role in defining

the goals of SPI and planning, managing, and controlling the SPI initiatives. In the agile software development context, on the contrary, the experience and knowledge of software developers and the self-organization of software developers in improving and adapting their daily working practices have been clearly placed in a central role.

In the agile approach, the role of management is to organize and co-ordinate rather than plan, execute, or control. Furthermore, the contextual needs for improving and adapting processes throughout the entire development process are emphasized, rather than the organizational goals in the regular SPI activities of development teams [Salo 2007].

The strong difference between the traditional and agile SPI could be seen when Salo (2007) compares both approaches considering the SPI element organizational models of continuous improvement, these differences are shown in table 1. The agile view is entirely based on team behavior.

**Table 1: Organizational SPI Models.**

Characteristic	Plan-Driven View	Agile View
SPI Approach	Top-down	Bottom-up
Primary Goal	Organisational procedures for improving the organisational software process(es)	Adapting the process to the contextual needs of individual project teams
		Improving the effectiveness of individual project teams
SPI Control	Organisational control of SPI	Self-organisation of teams
Knowledge Transfer	Explicit knowledge: external knowledge capture and inert knowledge transfer to support a learning paradigm (Lycett et al. 2003)	Face-to-face communication Tacit knowledge: Establishing and updating project knowledge in the participants' heads rather than in documents. (Boehm & Turner 2005)
Basis for Improvement	Organisational Goals	Contextual needs
	Measurements	Experience and Learning of Software Developers Regular team reflections

### 3.2 Agile SPI Based on tools automation - Process Centered Software Engineering Environments

Gruhn (2002) states that software process denotes the set of all activities which are carried out in the context of a concrete software development project. It usually covers aspects of software development, quality management, configuration management and project management. The description of a software process is called a software process model. A software process model does not only describe the activities which have to be executed, but also the tools to be used, the types of

objects and documents to be created/manipulated and the roles of persons involved in software development. One of the key questions for supporting software processes is how software process models and software engineering environments are related or ó in other words ó how a supporting infrastructure can be derived from the information given in a software process model.

Software engineering environments (SEEs) are meant to support software development. Process-centered software engineering environments (PCSEEs) give up the notion of a predefined process model. Process models used in this way should define which software development activities are to be executed when and by whom, they should identify tools to be used and the format of documents to be created and/or manipulated.

Then, the process model is interpreted at runtime to identify which process behavior has to be supported. In concrete this means, that software developers are reminded of activities which have to be carried out, automatic activities are executed without human interaction and consistency between documents is enforced up to a certain level [Gruhn 2002].

Actually PCSEEs is closely related to the purpose of software process improvement. The findings gathered during process modeling and the results of process model analysis usually indicate how actual processes can be improved in order to contribute to improved process productivity and product quality [Gruhn 2002].

Four elements of PCSEE are important to understanding tools automation in this context these elements are [Gruhn 2002]:

**Workflow management:** The idea of workflow management is that process models are used as basis of real processes, i.e. real processes are governed on the basis of the underlying process models. A workflow engine checks process models and the actual process state in order to identify which activities can be executed and lets potential participants know about it.

**Automation:** The process modeling purpose of automation is closely related to the purpose of workflow management. While workflow management tries to automate the coordination between activities, the immediate automation of single activities is another purpose of process modeling. In the context of software processes, activities like checking in and out of configuration units, building releases, evaluating test set coverage etc. usually can be automated. If automation is a major goal, then at least the preconditions for activities to be automated have to be described in detail.

**Monitoring:** Another purpose of process modeling is to measure the deviation between what processes are supposed to look like and how they actually behave. Then, traces of real world processes can be used to identify what the

deviations are. Usually, process monitoring is not a self-contained purpose, but only a first step towards SPI or process automation.

**Deriving support for tool integration:** Based on the assumption that not all parts of software processes are suited for workflow management support and that different parts of software processes deserve different kinds of support and based on the experience that different software processes require different tools, the integration of which results in process specific challenges, a modest or even though worthwhile or purpose of process modeling is to identify the requirements for tool integration in the context of a concrete process model.

The first value of Agile Manifesto (2001): *Individuals and interactions over processes and tools*, lead the agile practitioners to give little importance to using tools. But, seven years later, one of the agile manifesto signatories, Brian Marick (2008) stated that the first agile value should be revised once that tools help teams to deal with non creativity tasks.

Many agile teams are adopting tools for automate their process. Beck (1999) states that teams must have one computer testing and integrating source code all time. Adopting the four elements of PCSEE purposed by Ghrun (2002) and seeking their use in agile in teams

**Workflow management:** Many Agile teams are using issues tracking tools to control the stories flow inside the teams [Abrahamsson et al 2002]. In this scenario, these tools are acting as workflow management tools.

**Automation:** Agile teams are using continuous integration servers and tests robots to maintain the high quality of the final product [Abrahamsson et al 2002]. When automating these parts, the team releases its members for doing creativity tasks while automated tasks ensure that there will not be occasional failures in the process.

**Monitoring:** Spreadsheets, Charts, Wiki are commons tools for monitoring the progress of the projects. But many teams are including process metrics like builds broken or tests cases wrote per story [Jackobsen 2009]. This is an example of tools monitoring not the project but the process as well.

**Deriving support for tool integration:** Agile teams usually use different tolls for performing different tasks [Abrahamsson et al 2002]. The integration of all this tools becomes painful for agile teams challenging them. No one of these elements where formally studied in agile SPI perspective.

### 3.3 Agile Assessment Methods or Nokia Test

Following the Saloos (2007) definition for SPI elements, this subsection refers about the element (2) processes and assessments. An urgent need has been recognized for a set of guidelines for agility-compatible standard process maturity assessments and

also for a set of standards for the acknowledgement of agile methods by lead assessors (Boehm & Turner 2005).

The agile assessment method has been suggested as providing a lightweight approach for assessment to identify and adopt the most suitable agile methods amongst the existing organizational practices [Pikkarainen & Passoja 2005]. Furthermore, techniques have been suggested for increasing the agility level of a software development team by assessing the current agility level against the defined agility goals [Lappo & Andrew 2004].

Thus, the current discussion of process assessments in the agile context does not so much address the certification or define the maturity of the organizational software development processes, but rather evaluates the purpose of adopting agile practices (Salo 2007).

In this context was emerged the Nokia Test [Vodde 2006]. In 2005, Bas Vodde was coaching teams at Nokia Networks in Finland and developed the first Nokia Test focused on agile practices. He had hundreds of teams and wanted a simple way to determine if each team was doing the basics.

It does not provide the secret sauce for hyper performing teams. However, it is the first line of the recipe for high performance. These tests were applied in scrum teams at OpenView Venture Partners and to their portfolio companies as the venture group does not expect good performance from Scrum teams without passing the Nokia Test. They are also very interested in predictability of release dates which is impossible without passing grades on the test [Sutherland 2008].

The test consists in eight questions about how the teams are adopting Scrum. The Nokia Test is in two parts. The first part consists of three questions asking if the team is doing Iterative Development. The questions are [Sutherland 2008]:

É Iterations must be timeboxed to less than 4 weeks?

É Software features must be tested and working at the end of each iteration?

É The iteration must start before specification is complete?

The next part of the test checks whether the team is doing Scrum, in view of the Nokia, understanding, the remaining five questions are:

É You know who the product owner is?

É There is a product backlog prioritized by business value?

É The product backlog has estimates created by the team?

É The team generates burndown charts and knows their velocity?

É There are no project managers (or anyone else) disrupting the work of the team?

In 2008 Jeff Sutherland developed a Nokia Test scoring system. Each person on the team takes a sheet of paper and prepares to score eight questions on a scale of 1-10 and teams must average their score. In 2009, a team question was added to the Nokia test about teams. Sutherland created four categories for Nokia Tests results and linked these averages with following revenues [Sutherland 2008]:

É ScrumBut (Average < 8,0) - revenue up 0-35%

É Pretty Good Scrum (8,0 <= Average < 9,0) ó revenue up 150% - 200%

É Good Scrum (9,0 <= Average < 10,0) - revenue up 300%

É Great Scrum (Average = 10) - annual revenue up 400%

Sutherland (2008) stated that in the initial evaluations the teams are average 4.0, but in six months their average up to 7,0 and the velocity is about 300% of initial velocity. The Return of Investment (ROI) is about 11000% on first year since that is relatively easy and cheap to change scrum process for better. Answering the Nokia Test provide to the team some idea about how much it is really doing Scrum.

### 3.4 Agile Assessment Methods ó Comparative Agility

Kenny Rubin & Mike Cohn (2007) launched the Comparative Agility Survey. The Comparative Agility assessment is based on a simple, but powerful concept, determine how good are on company compared to its competitors. Rubin & Cohn assume agile teams and organizations strive always to be better than their competition and their past selves.

Comparative Agility assessments present the results of a set of survey responses in comparison to some other set of responses. Using Comparative Agility it is possible to compare a team, project or organization to:

- The total set of collected responses; responses from organizations in the same industry;
- Responses from similar types of projects (such as commercial software, websites, and so on);
- Responses from projects with similar lengths of experience at becoming agile.

The approximately 100 questions of a Comparative Agility assessment are organized into seven dimensions and thirty-two characteristics. The seven dimensions represent broad classifications of changes to be expected of a team or organization as it becomes more agile. The seven dimensions are:

- teamwork
- requirements
- planning

- technical practices
- quality
- culture
- knowledge creation

Each dimension is made up of three to six characteristics and a set of questions is asked to assess a team's score on each characteristic. Questions are answered on a fivepoint Likert scale of:

- True
- More true than false
- Neither true nor false
- More false than true
- False

Through a combination of Dimensions, Characteristics, and individual Questions, a team or organization can see how they compare to other organizations, or to themselves. The score ranges start in -2 (worst scenario) until 2 (Best Scenário). The average Scenário shows the 0 (zero) grade. The study involving this survey was presented in Cohn (2009).

### 3.5 Agile and Standards Process

The compatibility of agile software development approaches with the existing standard process models is one SPI issue that has been addressed in agile literature [Salo 2007]. One central problem has been posed as follows: How do you merge agile, lightweight processes with standard industrial processes without either killing agility or undermining the years you have spent defining and refining your systems and software engineering process assets [Boehm & Turner 2005]?

Some agile proponents have argued that people willing to spend money on CMM® certification are less interested in the agile value proposition, while those needing agility for business reasons are less interested in getting CMM or ISO 9000 certification [Williams & Cockburn 2003]. Nevertheless, mature software organizations especially are concerned about how the adoption of agile processes will affect their assessment ratings [Boehm & Turner 2005].

It has been argued that the synergy [Paulk 2001] and philosophical compatibility [Reifer 2003] of XP and CMM® have been agreed upon among most of the leaders in the field. However, shortages also between the ISO and CMM requirements and agile methodologies have been reported, along with a lack of practices to support the commitment of management to the defined software development process, and also regarding the setting up and staffing of an independent quality assurance group [Vriens 2003].

in addition, the degree of documentation and the infrastructure required by current process standards for lower-level certification are issues of concern [Boehm & Turner 2005].

From 2003 onwards, however, the IEEE Standards Association has conducted agile standardization work in the IEEE 1648 working group<sup>1</sup> to establish and manage software development efforts using agile methods.

#### 4 Conclusion

SPI was extensively studied in traditional software development, on the subject of SPI in the context of agile software development there are several avenues of investigation to be explored in future research. Even researches related to SPI on agile environments are studied in another point of view and this is a complicating factor for studying agile SPI in the light of traditional and established methods in traditional SPI.

Looking for the six elements purposed by Salo (2007), there are several avenues of investigation to be explored in future research either by lack of direction from the subject to process improvement or the lack of research itself. These lacks of research will be explored in the two MSc dissertations that were initiated by this literature review, but also may help other researchers to identify what kind of research could be realized when software process improvement is related to agile.

---

<sup>1</sup><http://standards.ieee.org/board/nes/projects/1648.pdf>

#### References

- [Abrahamsson et al 2002] Abrahamsson, P., Salo, O., Ronkainen, J., Warsta, J.: Agile Software Development Methods Available in: <http://www.pss-europe.com/P478.pdf>, last access in 3/22/2010. (2002)
- [Agile Manifesto 2001] Agile Alliance: Agile Manifesto for Software Development: Available in [www.agilemanifesto.org](http://www.agilemanifesto.org), last access in 03/22/2010.
- [Bach 1994] Bach, J.: The Immaturity of CMM, American Programmer Magazine Vol 3, pp 7--8 (1994)
- [Basili & Caldiera 1994] Basili, V. R., Caldiera, G.: Experience Factory. In: Encyclopedia of Software Engineering. Marciniak, J. J. (ed.). John Wiley & Sons, Inc. 469--476 (1994)
- [Beck, 1999] Beck, K. Extreme Programming Explained ó Embrace Change. Addison-Wesley. (1999)

[Boehm & Turner 2003] Boehm, B. & Turner, R.: Using Risk to Balance Agile and Plan- Driven Methods. Computer, Vol. 36, 6 (6), June, pp. 57-66, (2003)

[Boehm & Turner 2005] Boehm, B. & Turner, R.: Management Challenges to Implementing Agile Processes in Traditional Development Organizations. IEEE Software, Vol. 22(5), September-October, pp. 30-39, (2005)

[Cohn 2009] Cohn, M.: Succeeding with Agile: Software Development using Scrum. Addison Wesley Longman, Inc. 504 p. (2009)

[Deming 1990] Deming, W. E.: Out of the Crisis. 10 Printing ed. Massachusetts Institute of Technology, Center for Advanced Engineering Study. Cambridge (1990)

[Debou 1999] Debou, C.: Goal-Based Software Process Improvement Planning. In: Better software practice for business benefit: Principles and experience. Messnarz, R. & Tully, C. (ed.) IEEE Computer Society, pp 107--150, Los Alamitos, CA. (1999)

[Dybå 2000] Dybå, T.: An Instrument for Measuring the Key Factors of Success in Software Process Improvement. Empirical Software Engineering, Vol. 5, 357-390 (2000)

[Florac et al 2000] Florac, W. A., Carleton, A. D. & Barnard, J. R.: Statistical Process Control: Analyzing a Space Shuttle Onboard Software Process. IEEE Software, Vol. 17, 4 (4). July, August, pp. 97-106 (2000)

[Georgiadou 2004] Goergiadou E.: Software Process and Product Quality Assurance: A historical Perspective. Cybernetics and Systems Analysis. 11(4), pp 19--27 (2004)

[Goldensen & Herbsleb 1995] Goldensen, D. R., Herbsleb, J. D.: After the Appraisal: A Systematic Survey of Process Improvement, Its Benefits, and Factors that Influence Success. CMU/SEI-95-TR-009. Software Engineering Institute. Pittsburgh. (1995)

[Gruhn 2002] Gruhn, V.: Process-Centered Software Engineering Environments A Brief History and Future Challenges. In: Annals of Software Engineering, 14, pp 363--382,(2002)

[Herbsleb et al 1994] Herbsleb, J., Carleton, A., Rozum, J., Siegel, J., Zubrow, D.: Benefits of CMM-based software process improvement: Initial results. Technical report, CMU/SEI- 94-TR-13 (1994)

[Humphrey 1995] Humphrey, W. S.: A Discipline for Software Engineering. Addison Wesley Longman, Inc. 242 p. (1995)

[Jackobsen 2009] Jackobsen, C.: Scrum and CMMI: from Good to Great - are you ready - ready to be done ó done. In Proceedings of Agile Conference 2009. (2009)

- [Jarvinen 1994] Jarvinen, J.: On comparing process assessment results: BOOTSTRAP and CMM. In Software Quality Management, SQM94, pp 247--261, Edinburgh (1994)
- [Kitchenham 1996] Kitchenham, B., Pfleeger, S. L.: Software quality: The elusive target. In IEEE Software, 13(1), pp 12--21. (1996)
- [Krasner 1999] Krasner, H.: The Payoff for Software Process Improvement: What it is and How to Get it. In: Elements of Software Process Assessment & Improvement. El Emam, K.& Madhavji, N. H. (ed.). IEEE Computer Society, PP 113-130, Los Alamitos, California(1999)
- [Lappo & Andrew 2004] Lappo, P. & Andrew, H. C. T.: Assessing Agility. In: The proceedings of the Extreme Programming and Agile Processes in Software Engineering. June, 2004. Garmisch-Partenkirchen, Germany. Eckstein, J. & Baumeister, H. (ed.). Springer. pp. 331--338. (2004)
- [Lycett et al, 2003] Lycett, M.; Macredie, R. D.; Patel, C.; Paul, R. J. Migrating Agile Methods to Standardized Development Practice. In: Computer, pp. 79-85. IEEE Computer Society. Jun. 2003
- [Marick 2008] Marick, B.: Seven Years Later: What the Agile Manifesto Left Out. In: Agile Development Practices (2008)
- [Paulk 2001] Paulk, M. C.: Extreme Programming from a CMM Perspective. Software, Vol. 18, 6 (6), Nov. Dec, pp. 19--26. (2001)
- [Pikkarainen & Passoja 2005] Pikkarainen, M. & Passoja, U.: An Approach for Assessing Suitability of Agile Solutions: A Case Study. In: The proceedings of the Sixth International Conference on Extreme Programming and Agile Processes in Software Engineering. Sheffield University, UK. pp. 171--179. (2005).
- [Reifer 2003] Reifer, D. J.: XP and the CMM. IEEE Software, Vol. 20, 3 (3), May/June, pp. 14--15. (2003)
- [Rico 2004] Rico, D. F.: ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers. J. Ross Publishing. Florida, U.S.A. (2004)
- [Rubin & Cohn] Rubin, K., Cohn, M.: Comparative Agility Survey. Available in <http://www.comparativeagility.com/>, last visit in 03/22/2010.
- [Salo 2007] Salo, O.: Enabling Software Process Improvement in Agile Software Development Teams and Organisations, PhD Thesis, Oulu, (2007)
- [SEI 2006] Software Engineering Institute.: Capability Maturity Model® Integration (CMMISM), Version 1.2. Carnegie Mellon Software Engineering Institute. (2006)
- [Stamelos & Sfetsos 2007] Stamelos, I., Sfetsos, P.: Agile Software Development Quality Assurance. Information Science Reference, (2007)

[Sutnerland 2008] Sutnerland, J.: Money for nothing: And your change for free ó Agile Contracts. In Proceedings of Agile Development Conference (2008)

[Van Soligen & Berghout 1999] Van Solingen, R., Berghout, E.: The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development. The McGraw-Hill Companies. (1999)

[Vodde 2006] Vodde, B.: Nokia Networks and Agile Development. In Proceedings of EuroMicro Conference. (2006)

[Vriens 2003] Vriens, C.: Certifying for CMM Level 2 and ISO9001 with XP@Scrum. In: The proceedings of the Agile Development Conference (ADC03). September, 2003. IEEE Computer Society. pp. 120--124. (2003)

[Williams & Cockburn 2003] Williams, L. & Cockburn, A.: Agile Software Development It's about Feedback and Change. IEEE Computer Society, Vol. 36, 6 (6), June, pp. 39-43.(2003)

## Using Function Points in Agile Projects

- Célio Santana<sup>1,2</sup>, Fabiana Leoneo<sup>2</sup>, Alexandre Vasconcelos<sup>2</sup>, Cristine Gusmão<sup>3</sup>  
<sup>1</sup> Universidade Federal de Rural de Pernambuco, Unidade Acadêmica de Garanhuns, Av. Av. Bom Pastor S/N Garanhuns/PE, Pernambuco, Brazil CEP CEP 55.296-901  
<sup>2</sup> Universidade Federal de Pernambuco, Centro de Informática, Prof Luiz Freire Avenue. S/N,50740-540 Recife, Pernambuco, Brazil  
<sup>3</sup> Universidade Federal de Pernambuco, Centro de Telesaúde, Hospital das Clínicas | 2º andar | Av. Prof. Moraes Rego s/n , Recife , Pernambuco, Brazil CEP 50.670-420

{casj, fls2, amlv}@cin.ufpe.br, cristine.gusmao@nutes.ufpe.br

**Abstract.** Agile development has become increasingly common in the organizational software development environment. This paper examines whether function points would be compatible with story points on agile projects. Specifically, it addresses the question of whether function points are a relevant measure of velocity. Although any unit of measure can be used, this paper contrasts theoretical concepts about Story Points (SP) and function points (FP) as units for measuring size. It was also realized a statistical correlation between FP and SP using 2191 stories and 18 iterations in a Brazilian public agency. The conclusion drawn from this study was that function points, in that particular case, could be related with the initial value of the Story Points found after the planning poker.

**Keywords:** Function Point, Function Point Analysis, Story Points.

### 1 Introduction

The software industry is almost 60 years old, which makes it a fairly maturity industry. One would think that after six decades the software industry would have well established methods for measuring productivity and quality, and would also have collected a large volume of accurate benchmark data of thousand of measured projects. However, this is not quite the case [1].

Initially, to measure productivity and quality was used one unit called lines of code (LOC). At the time, circa 1950, that metric was fairly effective, once that coding took about 50% of the effort to build an application [1]. Between 1957 and 1967 the situation changed dramatically. Low level assembly languages started to be replaced by more powerful procedural languages such as COBOL and FORTRAM. Applications sizes grew from 1.000 lines of code past to 100.000 lines of code, raising problems when using LOC metrics [1].

These economics problems were what caused IBM to assign Allan Albrecht to develop a useful metric that was independent of code volumes, and which could both economic productivity and quality without distortion. In 1979, Allan Albrecht was the first to publish a method for functionally sizing software called function point analysis (FPA) [2].

The use of FPA, as a measure of the functional size of software, has grown since the mid 70s from a few interested organizations to an impressive list of organizations worldwide. The successful adoption of FPA was ratified with the creation of the ISO/IEC 14143:2007 [3].

Principles based in responding to changes quickly and light documentation. This vision seems antagonist than that proposed by traditional engineering [4]. Schuh [5] defines agile development as a counter movement to 30 years of increasingly heavy-handed processes, meant to refashion computer programming into software engineering, rendering it as manageable and predictable as any other engineering discipline.

Mnkandla [4] states that the agile movement could mark the emergence of a new engineering discipline, that has shifted the values of the software development

process from the mechanistic (i.e., driven by process and rules of science) to the organic (i.e., driven by softer issues of people and their interactions). Boehm [5] view believes in agile methodologies as a challenge to the mainstream software development community that presents a counter-culture movement, which addresses change from a radically different perspective.

To an agile team, a plan is one view of the future, but many alternatives are possible. As a team gains knowledge and experience, it will count into the plan [6]. A measurement system which supports this kind of experience is considered "special" and some metrics used in this context are Story Points (SP) and Ideal Days [1]. Jones [1] states that one of the agile weaknesses is the widespread failure to measure projects using standards metrics, such as function points. Based in this statement an ad-hoc search were conducted in the following sources: ACM Digital Library, CiteSeerX, IEEE Xplore, Scopus and SpringerLink.

Just one source was published, and presented relevant work showing scientific evidence about the function point analysis and story points running in agile software development environments. The work was published by Fuqua [7] who conducted a study by using function points in agile projects and tried to correlate with story points on those projects.

Other relevant work about this subject, not found in the ad-hoc search, was presented by Jones [1] in a book, where he states that according to its empirical basis, it is noted that two function points is equal to one story point on average. But it is worth noting that this measure is an average of its empirical database.

In this light, this paper presents conceptual differences between function points and story points. The goal of this paper is to present the theoretical relationship between story points and function points as well as providing empirical data from a real life case study where one project was measured using these two approaches.

After this introductory section, the section 2 explores function points analysis. This section 3 presents the concept of story points. Section 4 shows the size measurement concept which are related with FPA and SP. The section 5 shows theoretical differences between both techniques. Section 6 presents the case studied in a Brazilian public agency. And section 7 shows the summary, related works and threats of validity of the work.

## 2 Function Point Analysis

Once there was growth in the use of function points, there has been wider application and use of the measure. Since happened its formation in 1986, the International Function Point Users Group (IFPUG) has continuously enhanced the original Albrecht method for functionally sizing software.

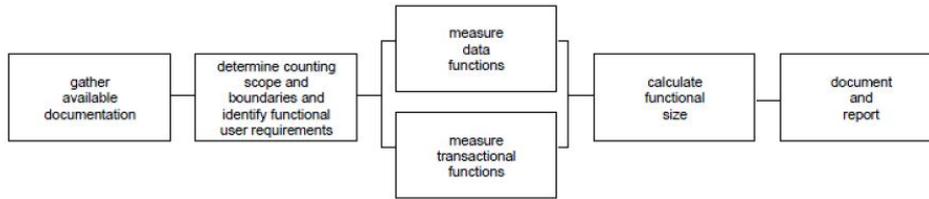
This International Standard is the latest release in the continually improvement IFPUG method. This aims to promote the consistent interpretation of functional size measurement in conformance with ISO/IEC 14143-1:2007. The IFPUG functional size measurement method is known as function point analysis and its units of functional size are called Function Points. The IFPUG version of Function Points is published in The Counting Practices Manual in its actual version 4.3 [8].

IFPUG's method for function point analysis is an ISO standard and must be conformant to ISO/IEC 14143-1:2007. The method can measure "functional size" not addressing "non-functional size". This does not mean that the nonfunctional size cannot, or should not, be measured. Instead it, must be clearly stated as a separate measure [8]. The process diagram of IFPUG FPA counting is shown in Figure 1.

The first stage in the function point counting procedure is to gather the available documentation. To support a functional size measurement, it shall describe the functionality delivered by the software or the functionality that is impacted by the software project that is being measured.

Suitable documentation may include requirements, data/object models, class diagrams, data flow diagrams, use cases, procedural descriptions, report layouts, screen layouts, user manuals and other software development artifacts. If sufficient documentation is not available, it is important to access experts who are able to provide additional information to

address any gaps in the documentation. The ideal requirements are called perfect requirements by the FPA practitioners.



**Figure 1. FPA Procedure Diagram [8].**

The next stage is responsible for counting scope which defines the set of Functional User Requirements to be included in the function point count. Also in this stage it is necessary to determine the boundary which is a conceptual interface between the software under construction and its users. Identifying the functional requirements is related with the concept of elementary process which is the smallest unit of activity that is meaningful to the user. To identify each elementary process, the following activities shall be performed:

- Is meaningful to the user;
- Constitutes a complete transaction;
- Is self-contained;
- Leaves the business of the application being counted in a consistent state,

Following these rules it is necessary to identify all unique elementary processes. After that, it is necessary to measure data functions, which represents functionality provided to the user to meet internal and external data storage requirements. A data function is either an internal logical file or an external interface file. An internal logical file (ILF) is a user recognizable group of logically related data or control information maintained within the boundary of the application being measured.

An external interface file (EIF) is a user recognizable group of logically related data or control information, which is referenced by the application being measured, but which is maintained within the boundary of another application. In parallel, could be performed the measuring of transactional functions. A transactional function is an elementary process that provides functionality to the user for processing data. A transactional function is an external input, external output, or external inquiry.

An external input (EI) is an elementary process that processes data or control information sent from outside the boundary. The primary intent of an EI is to maintain one or more ILFs and/or to alter the behavior of the system. An external output (EO) is an elementary process that sends data or control information outside the application's boundary and includes additional processing beyond that of an external inquiry. The primary intent of an external output is to present information to a user through processing logic other than or in addition to the retrieval of data or control information. The processing logic must contain at least one mathematical formula or calculation, create derived data, maintain one or more ILFs, and/or alter the behavior of the system.

An external inquiry (EQ) is an elementary process that sends data or control information outside the boundary. The primary intent of an external inquiry is to present information to a user through the retrieval of data or control information. The processing logic contains no mathematical formula or calculation, and creates no derived data. No ILF is maintained during the processing, nor is the behavior of the system altered.

On systems which present perfect requirements, different measurements performed by different people must have the same final result. Imperfect requirements lead the measurement performer to take assumptions about that requirement, and this kind of assumption could lead to different results in counting the same sample of requirements. Because of this property, it is considered that FPA is an objective method for measuring software.

### 3 Story Points

Story points are a unit of measure for expressing the overall size of a user story, feature, or other piece of work. When we estimate with story points we assign a point value to each item. The raw value we assign is unimportant. What matters are the relative values [6].

The number of story points associated with a story represents the overall size of the story. There is no set formula for defining the size of a story. Rather, a story point estimate is an amalgamation of the amount of effort involved in developing the feature, the complexity of developing it, the risk inherent in it, and so on [6].

There are two common ways to get started. The first approach is to select a story that you expect to be one of the smallest stories you will work with, and say that story is estimated at 1 story point. The second approach is to select a story that seems somewhat medium-sized and give it a number somewhere in the middle of the range you expect to use. A story that is assigned a two should be twice as much as a story that is assigned a one [6].

Story estimates need to be owned collectively by the team. A story comprises multiple tasks and a task estimate is owned by the individual who will perform the task. Story estimates, however, are owned by the team for two reasons: First, since the team do not know yet who will work on the story, the ownership of the story cannot be more precisely assigned than to the team collectively. Second, estimates derived by the team, rather than a single individual, are probably more useful [9].

At the end of an iteration the team counts the number of story points they completed. Then they use that as a forecast of how many story points they'll complete in upcoming iterations of the same length. The term velocity refers to the number of story points a team completes, or expects to complete, in an iteration [9].

### 4 Size Measurement

A software measurement is a quantifiable dimension, attribute, or amount of any aspect of a software program, product, or process. It is the raw data which are associated with various elements of the software process and product. A typical set of metrics might include [10]:

- Quality.
- Size. (target of the study)
- Complexity.
- Effort
- Productivity,
- Cost.
- Schedule.
- Rework.

Two ways for measuring software size were catalogued in 1992 [11]. The first consider the physical source lines and logical source statements. Counts of physical lines described size in terms of the physical length of the code as it appears when printed for people to read.

The other way counts of logical statements, on the other hand, attempt to characterize size in terms of the number of software instructions, irrespective of their relationship to the physical formats in which they appear. Both function points and story points measure the size of the software and are based on the count of logical expressions.

Function points address functional size [8] while story points represent the business value of one user story [6]. In fact, Agile teams separate estimates of size from estimates of duration [9] while function points are complemented by other methods when it comes to effort and cost estimate such as COCOMO II [12].

## 5 Function Points x Story Points

Although FP and SP estimate the size of the software to be delivered, some particularities make measures applied by both techniques to the same product have different sizes, variations and deviant behavior at the end of the measurement. Some of the strongest of these particularities are detailed in the following subsections.

### 5.1 Team expertise X Standardized Methods.

A nice feature of story points is that each team defines them as they see fit. One team may decide to define a story point as an ideal day of work and another team may define a story point as a measure of the complexity of the story [9].

In the last statement, Cohn suggests that the story points can vary from several teams based in their experience to assess the effort, complexity and risk associated with certain stories. Any assumptions made in function points are considered a counting interpretation.

A specification bringing perfect requirements, where no assumptions are made, must present the same final result. Any assumptions regarding primary intent must be documented for helps in next counting. Thus function point leaves no space for using expertise.

For Example, considering function point a small function for including one email address in a virtual schedule may be the same size of a function with perform a complex integral calculus with receive one equation as parameter and return the string with the result. The same example in story points should present very different results and these results could be different among different teams.

In the other hand, considering function points, store one formulary containing fifteen fields may be different for including one containing sixteen, while in story points this kind of difference is rare. So this aspect is seen in a different way in both techniques.

### 5.2 Functional Size x Product Size.

According to ISO 9126 [13] non-functional requirements are that specifies criteria which can be used to judge the operation of a system, rather than specific behaviors. This should be contrasted with functional requirements that define specific behavior or functions.

Considering the IFPUG definition, functional size is the size of the software derived by quantifying the Functional User Requirements we should assume that nonfunctional requirements are not covered in function points [8]. The IFPUG Framework for Functional Sizing [14] defines some kind of sizes presented in the software development such as functional size, technical size and quality size which are related to:

- **Functional User Requirements:** a sub-set of the user requirements. The Functional User Requirements represent the user practices and procedures that the software must perform to fulfill the users' needs. They exclude Quality Requirements and any Technical Requirements
- **Quality Requirements:** any requirements relating to software quality as defined in ISO 9126:1991
- **Technical Requirements:** requirements related to the technology and environment, for the development, maintenance, support and execution of the software.

The combination of the functional size, technical size and quality size represents the Product Size. But, this concept is not detailed by the IFPUG. Considering the statement that story point estimate is an amalgamation of the amount of effort involved in developing the feature, the complexity of developing it, and the risk inherent in building it [6], it looks like the story point is concerned in defining a product size since the agile team considers any kind of risk and complexity to determine the size of the story, and these assumptions are related to Quality and Technical requirements.

Nowadays, IFPUG is building a metric called Software Non-Functional Assessment Process (SNAP). The SNAP Project Team expects to develop a project assessment method that will use series of questions grouped by category to measure the impact of non-functional requirements on the development and delivery (size) of the software product. The resulting size will be the size of the non-functional requirements, just as the functional size is the size of the functional requirements [15].

In a simple way, we still cannot consider the theoretical concept that  $SNAP\ size + FP\ Size = SP\ Size$  because the agile method considers the environment of the project and not just the product. For example, in function points a bookstore which have no requirements for security, available, performance and its access are made in a local machine, will have the same FP size of this same bookstore considering the same restrictions of the amazon.com. In story points, the amazon.com will be much larger than its offline, unsecure, slow and unstable version.

### 5.3 Small Pieces x Whole Product.

In the Agile Manifesto were defined 12 principles, which one of them states: "Deliver working software frequently, from a couple of weeks to a couple of months, with preference to the shorter timescale." This statement reinforces the adoption of interactive life cycle, largely adopted in agile projects.

This continuous delivery in small "timeboxes" reduces the total of points delivered in iteration. Sometimes one big story, called epic, must be disaggregated for fit in one cycle. In fact, split stories are not a simple task in agile projects. There are number of times when it may be necessary to split a user story into multiple smaller parts [6]. First, a user story should be split when it is too large to fit within a single iteration. Sometimes a user story won't fit in iteration because it is bigger than a full iteration.

Alternatively, a story may be small enough to fit within iteration but it won't fit within the iteration being planned because there isn't enough room left. The team may feel they will have time to develop a portion of a story in the iteration, but not the entire story. Second, it can be useful to split a large user story (an epic) if a more accurate estimate is necessary.

But the question about splitting stories is raised from another Cohn's statement [9]: When a story, possibly an epic, is disaggregated into its constituent stories, the sum of the estimates for the individual stories does not need to equal the estimate of the initial story or epic. Similarly, a story may be disaggregated into constituent tasks. The sum of the estimates for the tasks does not need to equal the estimate of the initial story.

Thus, splitting stories seems to be a team decision and there are no rules about how to split and how distribute the points, making this disaggregation a particular process that works only for that team in that environment. Looking for function points splitting is not a problem. No data function or transactional function should be broken because they must follow the elementary process definition: "smallest unit of activity that is meaningful to the user."

Even if a function must be broken for a technical reason, it only will be considered complete when all of the functionality is completely developed, which means delivered zero function points or all function points to the user. But an anomalous behavior can be seen in the use of function points, in interactive and incremental projects, if the boundary of the counting just considers what is delivered in each iteration. In this case, the sums of the parts are bigger than the whole.

For example, a particular product is being built in an interactive and incremental whose two iterations have already been completed. In first iteration four features were delivered totaling fifty function points. In the second iteration another four features were delivered, but one of these feature (and already counted with 10 function points) in first iteration were updated for technical reasons totaling sixty function points delivered in second iteration. The first iteration + second iteration a hundred points, but that function which was built in first iteration and updated in the second, was counted twice and this just happens because the boundary of the counting is not the whole product, and which are delivered in each iteration. In function points the sum of the parts could be bigger than the whole (never smaller).

This problem does not occur in story points because the cost, in points, for updating one feature is embedded in the original story.

#### 5.4 Maintenance and Changes

According to ISO/IEC 14764 identified four categories of maintenance [16]:

- Corrective maintenance: Reactive modification of a software product performed after delivery to correct discovered problems.
- Adaptive maintenance: Modification of a software product performed after delivery to keep a software product usable in a changed or changing environment.
- Perfective maintenance: Modification of a software product after delivery to improve performance or maintainability.
- Preventive maintenance: Modification of a software product after delivery to detect and correct latent faults in the software product before they become effective faults.

Agile software development considers the corrective maintenance as a bug, and this kind of problem must not be managed, but solved. Which means that must not be sized. But adaptive maintenance (evolutionary maintenance), perfective maintenance and preventive maintenance (refactoring), are considered and evaluated in agile projects as new stories. When maintenance needs to be performed, a new history is written for that specific demand.

The functional size measurement quantifies the size of business requirements. In an enhancement environment, it measures the effects of changes to those business requirements. Therefore, functional size measurement is applicable to a subset of adaptive maintenance. This includes the software functionality added, changed or deleted as well as the software functionality provided to convert data and meet other implementation requirements [8].

Function points clearly do not fit the types of corrective, perfective and preventive maintenance, fitting only a few cases of adaptive maintenance. A project that has undergone many changes may have enhanced the difference in scores between the two approaches.

#### 5.5 One Requirement X Many Requirements

Measuring a single feature using the two techniques and compare their variation, maybe the most logical path to be taken when attempting to evaluate the relationship between the two methods. And repeat this process for the all others features of the project in an attempt to increase the historical basis, would be the next step in this comparison.

In terms of story points this idea may not be the best. Cohn [9] states that the central limit theorem tells us that the sum of a number of independent samples from any distribution is approximately normally distributed. For our purposes, this means that a team's story point estimates can be skewed way toward underestimation, way toward overestimation, or distributed in any other way.

But when we grab an iteration's worth of stories from any of those distributions, the stories we grab will be normally distributed. This means that we can use the measured velocity of one iteration to predict the velocity of future iterations. Naturally, the velocity of only one iteration is not a perfect predictor. Similarly, velocity may change as a team learns new technology, a new domain, or becomes accustomed to new team members or new ways of working. This means that to predict the behavior of the score, or the velocity, of a team is best to consider all stories delivered than each one individually.

Was also seen in the previous section that to consider the score of a feature function points can be misleading. The measure of the sum of the parts can be greater than the whole, thus sizing all features is more accurate than sizing one by one as well. The last reason to evaluate all the features is that there may not be a perfect match between a story and a requirement. So could be difficult to assign exactly which stories are equivalent to what requirements, and evaluating all iterations may bring a greater reliance to the comparison.

## 6 Case Study

The *Agência Estadual de Tecnologia da Informação do Estado de Pernambuco*, hereafter called ATI, is following the Brazilian Federal Government instruction known as the: *Instrução Normativa 04 de Maio de 2008*, hereafter called IN04, which came into force on 2 January 2009 the Department of Logistics and Technology of the Planning Ministry [17].

This instruction, in its article 14, states that the outsourcing strategy must define the understanding of the task to establish procedures and criteria for measurement of services provided, including metrics, indicators and values. With this technique Function Point Analysis has been adopted as currency in the local authority outsourcing contracts of software products.

ATI was forced to be adherent to this instruction in early 2010. Before that, since January 2009, ATI had been using Scrum as tool for contract management [17]. After the adoption of this instruction ATI continued to manage its suppliers through the scrum, but the payment of invoices should be measured based on the product delivered sized using function points.

ATI and its supplier held a planning poker meeting where it is sized in story points all demands of that sprint. But now it was necessary to conduct an estimative counting in function points required by IN04 for project planning. At the end of the sprint is still necessary to perform a counting in function points to determine the size of the product delivered and thus pay the suppliers.

The estimative counting is needed only for the allocation of project resources, which does not demand an accurate count but only an approximation of reality. But while ATI can count about 5000 story points per day, the ability to count function points is reduced to 600 function points in a day. And this estimative counting is considered *obscure* being unnecessary in most cases.

The record of the demands shown in Figure 2. The functional size and story points size of each demand is stored in this tool and can be recovered directly from the MySQL database which store the Mantis database.

Esforço Total Planejado (hs)	
Esforço Total Realizado (hs)	
SGNET - Data de Entrega da OS Prevista	
SGNET - Data de Entrega da OS Realizada	
SGNET - Tamanho	8 Story Point
SGNET - Tipo de Solicitação	Manutenção Evolutiva
Tamanho Ponto de Função	6 Function Point
Arquivos Anexados	 Mantis - EST - Processo - Ocorrências funcionais(Estagiários) - (Cliente - ATI).doc [^] (75,7

Figure 2. Recorded data about

## 7 Conclusions

Despite the strong differences of size definition presented in function points and story points, were presented empirical evidence for a real life project realized by a Brazilian public agency, showing the correlation between functional size and number of story points which are delivered at the end of each sprint.

The strength of this correlation suggests a further distancing between the two variables studied which may come from the differences presented in section 4 of this work.

The result cannot be generalized, but it supports an idea that Product Size = Functional Size + Non-Functional Size + Environments Variables Size, Story Points = Function Points + Non-

Functional Size + Environments variables Size. This formula is not intended to be shown mathematically correct, but that represent that the functional size is part of the product size and finding a correlation between the whole (product size) and the piece (functional size) represents a valid proportion.

Obviously is necessary to respect the units of measurement and the reality of each organization, so the result itself is not valuable, but the method of assessment, if replicated in more environments, may prove be useful for a particular company.

### 7.1 Discussion of results

Even being used to the same goal, function points and story points presents strong theoretical differences. Whereas the results of this study it is still surprising, once was observed a correlation between the functional size that is obtained accurately with impersonal method of sizing, and story points obtained purely from the experience of the team.

Especially if we evaluate this short history about this subject, starting with the Fuqua's work [7] where he performed a correlation between function points and a set of indicators used by your company, then performed the same correlation using story points and found no significant correlation between these two variables.

Although this study [7] used another function points, now known as mark II [20] which is different from official function points provided by the IFPUG. The basis of impartiality in Mark II counting method remains. The work of Jones [1] only presents the statement of the relationship between  $FP \setminus SP = 2$  without more information on how the result was obtained. It is obvious that the intention of this work was not to generalize their results, but was expected to find the same results than Fuqua [7] and close it as more empirical evidence strengthening the argument toward to "not fit". Facing the expectations and the results, we believe that the statement raised in the Framework for Functional Sizing of IFPUG [14] where it states that product size is a combination of (quality size, technical size and functional size) or  $Product\ Size = Non\text{-}Functional\ Size + Environments\ Variables\ Size + Functional\ Size$ .

Of course (x) that different companies presents different "sizes" for their store points and different proportion of the impact of functional size into the product size, but the goal of this paper is to motivate how these companies can find their ratio between FP and SP.

### 7.2 Implications for Research and Practice

The implications of this study for practice in first place concerns to ATI itself, and the possibility for perform a linear correlation for find a conversion method between story points and function points. Another practical implication is the description of a method that can be used by companies that are facing the same problem of the ATI and need a solution about how to assess the relationship between FP and SP within their organization. Remembering

that the values found in this work will be only valid for that ATI project, but organizations can use this method in its own database and so find their own correlation.

Those that are successful, including ATI, can perform a linear regression and find a first degree equation ( $y = Ax + B$ ) where y refers to the number of function points, is the amount of story points and A, B are constants. From this equation, companies can predict with a certain margin of error, which is the value of these variables from one another.

The first implication of this work in research is to present other empirical results joining a small base of scientific information about the subject and the first empirical study, not considering Jones's work [1], adopting IFPUG method. Other implication is formally present the main differences between the two approaches in section four. Surely, there is plenty of theoretical information compiled about the subject. But still is the possibility for gathering current data from systematic reviews or systematic mapping as well as creating data from new experiments and case studies that will enrich the knowledge of the academy.

In order to present the idea that story points are related to product size and function

points, or functional size, is only part of the product size. What seems clear is that the proportion between functional size and product size is different in every environment and can even be irregular within the same project.

### 7.3 Threats to Validity

The first group of threat validity stems from a lack of theoretical concepts consolidated about a possible correlation between the approaches. This fact may have contributed to weaken several factors in this study such as the wrong selection of the method or the pooling of demands. It is an "exploratory" study which portrays a more specific need than a company that intends to conduct scientific research. Indeed, this threat do not touch the section 5 of this work that could bring contributions if we were free of the limitations that are in the following section.

Another threat that comes from this factor is the lack of information about a demand that could help in their treatment. For example there is no way of knowing whether a demand is perfective or corrective maintenance (which could be dropped from the study because function points do not support them) or whether it is adaptive maintenance. Another problem with this group is that the number of samples (18) are still small to reach any definitive conclusion on this study. Finally the latest threats come from the validity of the statistical method used in this work. The lack of knowledge prevents to determine which type of method is most appropriate for the conduct of case studies and experiment. For example, the statistical method used in this work and the work Fuqua [7] were different. In this method the assumption of measuring the set of demands instead one individual (section 5.5) can bring bias to this study.

### 7.4 Limitations

The main limitation is the small amount of professionals who knows well the two techniques involved in this study. The impact of this work is the small number of sprints that could be counted because it was inexistent the counting of function points from September until December, which would be four more sprints for data collection. The second limitation was not performing the linear regression to support with more strength the results of the work, although with a rank correlation of 0.71 and a high confidence interval is very difficult that there is not a valid linear regression for this correlation.

### 7.5 Future Work

In industry, one future work is suggested that is the discovery the first degree equation  $FP = (x * SP + y)$  where FP is equal to the total function points delivered after Sprint, SP represents the estimate given by points in history, x and y are constants. This time it included a regression analysis to identify the function conversion between the variables.

To academy we present, as future work, the attainment of studies using formal secondary collect data method such as systematic review or systematic mapping on the relationship between function points and story points. Another future work is providing more empirical information about the relationship between FP and SP, to confirm the relationship  $Product\ Size = Functional\ Size + Non-Functional\ Size + Environments\ Variables\ Size$ .

### Acknowledgments

Célio Santana is a doctoral student at the Center of Informatics of the Federal University of Pernambuco where he receives the funding from the Brazilian National Research Council (CNPq), process #141156/2010-4. The authors thank CNPq for partially funding the participation in the XP@2011 Conference.

## References

1. Jones, C.: Applied Software Measurement, McGraw Hill. (2008)
2. Albrecht, A. J.: Measuring Application Development Productivity. In Proc. IBM Applications Development Symposium. GUIDE Int and Share Inc., IBM Corp., Monterey, pp. 83. (1979)
3. ISO/IEC 14143-1:2007, Information technology ó Software measurement ó Functional size measurement (2007)
4. Mnkandla, E., Dwolatzky, B.: Balancing the human and the engineering factors in software development. Proceeding of the IEEE AFRICON 2004 Conference, pp. 1207--1210, (2004).
5. Boehm, B., Turner, R.: Balancing agility and discipline: A guide for the perplexed 1st ed. pp. pp. 165-194, Addison Wesley (2004).
6. Cohn, M.: Agile Estimating and Planning. Addison-Wesley. (2005)
7. Fuqua, A.: Using Function Points in XP ó Considerations. In Proceedings of Extreme Programming and Agile Processes in Software Engineering, Springerlink.
8. IFPUG: Counting Practices Manual V. 4.3. Available in [www.ifpug.org](http://www.ifpug.org). (2009)
9. Cohn, M.: User Stories Applied. Addison-Wesley. (2004)
10. Glass, R. L.: Building Quality Software. Prentice-Hall. (1992)
11. Park, R. E.: Software Size Measurement. A Framework for Counting Source Statements. SEI technical report available in: <http://www.sei.cmu.edu/reports/92tr020.pdf>. Last Access in 03/01/2010.
12. Boehm, B., Abts, C., Brown, A. W., Chulani, S., Bradford, K. C., Horowitz, E. Madachy, R., Reifer, J., Steece, B.: Software cost estimation with COCOMO II. Englewood Cliffs, NJ:Prentice-Hall. (2000)
13. ISO/IEC 9126-1: Software engineering ó Product quality ó Part 1: Quality model. (2001)
14. IFPUG: Framework for Functional Sizing. Available in [www.ifpug.org](http://www.ifpug.org). (2003)
15. IFPUG: Software Non-functional Assessment Process. Available in [http://www.ifpug.org/about/SWNNon-FunctionalAssessmentProcess\\_Final20-2009.pdf](http://www.ifpug.org/about/SWNNon-FunctionalAssessmentProcess_Final20-2009.pdf) . Last Visit in 03/01/2011 (2009)
16. ISO/IEC 14764: Software Engineering ó Software Life Cycle Processes ó Maintenance (2006)
17. Department of Logistics and Technology.: *INSTRUÇÃO NORMATIVA Nº 4, 19 de maio de 2008*. Available in: <http://www.governoeletronico.gov.br/anexos/instrucaonormativa-n-04>. Last Access in 30/08/2010. (2008)
18. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall. (2004).
19. DeGroot, M., Schervish M. J.: Probability and Statistics 3 edition, Addison Wesley. (2001)
20. United Kingdom Software Metrics Association (UKSMA): "MKII Function Point Analysis Counting Practices Manual", Version 1.31 (Mk II FPA). Online at <http://www.uksma.co.uk> (1998)

## The Confidence of Agile Assessment Methods in the Context of Software Process Improvement

Célio Santana

Universidade Federal Rural de Pernambuco, Unidade Acadêmica de Garanhuns. Garanhuns. Brazil.

[Celio.santana@gmail.com](mailto:Celio.santana@gmail.com)

Cristine Gusmão

Universidade Federal de Pernambuco, Núcleo de Telesaúde (NUTES). Recife, Brazil.

[cristine.gusmao@nutes.ufpe.br](mailto:cristine.gusmao@nutes.ufpe.br)

Danilo Caetano

Universidade Federal de Pernambuco, Centro de Informática (Cin). Recife, Brazil.

[dcms@cin.ufpe.br](mailto:dcms@cin.ufpe.br)

Alexandre Vasconcelos

Universidade Federal de Pernambuco, Centro de Informática (Cin). Recife, Brazil.

[amlv@cin.ufpe.br](mailto:amlv@cin.ufpe.br)

**Abstract** with the advent of CMMI in the late 1990s, software companies tried to use a kind of score to rank themselves in the market. Being a CMMI level five company for a long time represented the epitome of highly qualified company. This phenomenon is also emerging in the agile software development with the emergence of methods such as Nokia or comparative test agility. This paper presents an analysis of how these methods are placed in the context of agile software development and its contribution to the improvement of software process in the same context from a case study developed in real companies.

**Keywords**-Agile Assessment Methods; Software Process Improvement; Agile Processes.

## Introduction

The success of the Capability Maturity Model for software, called from now SW-CMM, at the end of the 1990s was not only obtained by the proposed improvement of process maturity, but mainly because a particular company could assign a score to your own process.

This score was relative to the level of maturity that the process of an organization could achieve, which was the most immature level called the initial or also called level 1, and the more mature level known as the "Optimization" or Level 5 [1].

From then, any company could plan its process improvement toward SW-CMM maturity level five. This improvement could be achieved gradually since the division of the process in about twenty key process areas (KPA) and each could be improved in a manner almost independent of the others [1].

However, more than one mechanism for Improving the Software Process, the score of a process of a company in the CMM has become a criterion for classification. And the reference of this model was so strong that a company SW-CMM Level five was considered better than a SW-CMM Level three for almost all market segments without the need for other indicators [2].

For agile development to first draft evaluation was given by Paulk (2001) where an assessment was made of an agile method called Extreme Programming and its partial adherence to the SW-CMM model [3].

Using traditional models such as SW-CMM, lack the needed reference information for agile based Software development efficiency evaluation [4].

One problem is that, even if the traditional assessment is often seen as an opposite to the agile thinking, the agile software development should be based on the best software development practices. Simplifying does not mean not documented or not existence processes [1, 5].

Looking at the context of Software Process Improvement (SPI) where Salo (2007) states that agile software development provides new possibilities for conducting SPI, which may well provide grounds for meeting some of the central challenges of traditional SPI [6].

Agile software development provides a highly untraditional approach to SPI, in which the process improvement knowledge of software developers and software development teams is acknowledged and valued. In this context using traditional models of assessment could not consider these aspects [6].

Thus, Agile Assessment does not need to be a complex evaluation including the full analysis of CMMI base practices. It should be light-weight and based on agile principles, such as face-to-face communication, rapid feedback to interviewees and organization management and include the simple documentation [7].

The first documented agile assessment was given by Boehm and Turner (2003) [8] and since then other agile approaches to assessment have been suggested.

However, the agile community not seems interested in the idea of assign a score in the capacity of both individual and teams because this behavior decreases collaboration and increases competition [8].

The motivation of this research is to investigate the role of the Agile Assessment Models (AAS) considering the context of software process improvement in industry. Since there is no interest to determine a rating for comparative purposes, then the research question being asked is how these models can assist in improving the process in agile development.

The objective is to determine how the AAS Models help teams and organizations to improve their processes. To this end, case studies will be conducted to collect empirical evidence on real organizations using agile methods in their application.

After this introductory section, the section 2 explores SPI, the section 3 explores the AAS, the section 4 shows the research structure, in section 5 final considerations are exposed.

## SOFTWARE PROCESS IMPROVEMENT (SPI)

A software process can be defined as the sequence of steps required to develop or maintain software aiming at providing the technical and management framework for applying methods, tools, and people to the software task [9].

The SPI aims providing software development organizations with mechanisms for evaluating their existing processes, identifying possibilities for improving as well as implementing and evaluating the impact of improvements [10].

Traditionally, the ultimate goal of SPI in organizations is to provide a Return on Investment (ROI) for the organization through the improvement activities yielding more money than is spent on them [11].

ROI has been reported for various SPI achievements, such as improved efficiency of the development process and reduction of total software costs, increased quality of the end product, higher predictability of cost and schedule, and increased level of reuse [12].

The goal of the traditional software process is to provide high predictability, stability, and repeatability using highly managed and quantitatively monitored software development processes. On the other hand, agile principles highlight the need for the software process to be flexible, to be able to rapidly respond to the constant changes and context specific needs of software development [6].

As a result, traditional software development emphasizes up-front contract negotiations where the requirements, cost and schedule of the product development are fixed and the end product will be delivered at the end of the project lifecycle.

In this mode of software development, traditionally, extensive documentation and quantitative monitoring of the product development process plays a central role while. The principles and practices of agile software development, in turn, address the constant changes. [6].

It should be emphasized that Agile SPI still intended to improve organizational indicators and not just fit better to Scrum or XP or any other method chosen. That is, reduce bureaucracy although is an agile solution it is not the main result expected by Agile SPI.

## Agile Assessment Methods (AAS)

Boehm and Turner [8] present five factors that influence the performance of an agile team, these factors are critically, personnel, dynamism, culture and project size. Although the recognition of these factors is a good starting point for agility evaluation, the authors does not address any specifics regarding the application of an agile method. After this, the following AAS were proposed in chronological order.

### **Assessing Agility**

Lappo & Henry [13] in 2004 proposed the concept of measurable goal that is different from the metric itself. So when you set goals for the process and these targets are set by the company is perceived an improvement-oriented approach that does not rank the teams, but have quantitative targets from which to arrive.

### **Agile Assessment Approach**

Pikkarainen & Passoja [7] in 2005 proposed an agile approach to evaluate the goals using agile practices in agile or traditional projects. The Agile Assessment approach helps organizations to find the best suitable agile practices to improve a specific aspect of the software development work.

### **Nokia Test**

In 2005, Bas Vodde was coaching Scrum teams at Nokia in Finland and developed the Nokia test based on Scrum practices. The test was administered to teams in order to assess whether the team members were with the basic knowledge of Scrum consolidated [14].

Each staff person should get a piece of paper and prepare to calculate the score of the nine questions on a scale of zero to ten, and at the end, you should make a mean score [15].

### **Comparative Agility**

In 2007 Kenny Rubin and Mike Cohn proposed the Comparative Agility Survey which is based on a simple but powerful concept of determining how good a company is compared to its competitors. Rubin & Cohn assumed that agile teams and organizations are always looking to make the best compared to its competitors and its own past. The possible range score of this test is from -2 (worst) until 2 (best) score [16].

### **Others Approaches**

Other methods proposed in academia that were not deeply surveyed are: The Agile Hybrid Assessment Method for Automotive (AHAA) proposed in 2008 and used in automotive industry [17].

The Quantitative Agile Assessment [18], where each member of the team answer a questionnaire and one score are provided based on answers.

In the industry we can also find the Agile Assessment<sup>9</sup> provided by the thoughtworks.

## The research Structure

---

<sup>9</sup><http://www.agileassessments.com/>

The structure of the research is summarized in Table 1, where it is shown the methodological reference frame. The research is based on a case study to evaluate whether the AAS are linked to the results of organizational indicators. Other information about the research is showed in the following subsections.

## Premises

Since SPI is closely related to ROI, it is expected that when these improvements occur, they are reflected in the final results of an organization. This means that a change in the process should not only bring a greater suitability to the method used, but also increased productivity, profitability, return or any other social indicator that an organization should set as a goal.

TABLE I. METHODOLOGICAL FRAME

Methodological Reference Frame	
Approach Method	Inductive
Variables Nature	Quantitative
Method of procedure	Statistical
Objective	Descriptive and Causal
Scope	Field Study (Case Study)

This research will be taken as a premise that the improvements made in the way of developing software, when they bring a better result in ROI, will also bring an increase in the classification of AAS.

For example, a company that uses Scrum Nokia test evaluated by averaging 6.0, while performing an action of process improvement, not only will your Nokia Test average but also increase its productivity (or other indicator of ROI).

Different companies will present different measures, different ways for measuring, different goals and, of course, different results. Whereas any action of SPI is reflected in the ROI and also on adherence to agile approach used, it is possible to perform a correlation between the two indicators and assess the impact of change scores on the method and the result obtained by the company.

It is unlikely that changes in the way of developing software do not change the final results of the indicators of an organization, as well as their way of realizing the SCRUM or XP.

The use of multiple metrics is not, a priori, a bias because it is not being investigated the influence of a specific indicator, but a Boolean relationship between the outcome in AAS and the indicator of ROI. That is, if the evaluation scores of the assessment increase, the ROI also increase?

The AASs chosen are Nokia Test for scrum teams and Comparative Agility for XP hybrid scrum teams. These methods were chosen because they provide a final score, 0 to 10 in Nokia Test and -2 to 2 in Comparative Agility.

## Research Phases

The first stage of the research is the selection of companies where the case study will be conducted. From there, will be identified which ones use scrum and which ones use other methods at the same time will be also identified the key organizational indicators of each company for future comparisons.

The next phase is the preparation of forms that will be used in the data collections generated in each organization. This form must be unique for each company because it contains questions from the AAS chosen to that organization and its own indicators.

The next phase will be the creation of spreadsheets that will assist in analyzing the data received of each company individually, as well as the consolidated data.

## **Data Collection**

The forms provided to the companies will be divided in two parts: The first contain the AAS questions, the quantity this part of the form can vary since that Nokia Test and Comparative Agility is answered by all the team. The second part contains the result of the indicators chosen to that organization. The number of this part of the form can vary based in how much teams produces that indicator, or how many time the indicator is collected.

The monitoring of these companies will be longitudinal, and then the end of each cycle the companies must deliver the completed forms. However, companies presenting weekly cycle may not have significant changes in the outcome of AAS, which can result in the two parts of the form being delivered at different times. These cases will be considered baselines of time as may be monthly or bimonthly for companies who use longer cycles.

## **Data Analysis**

Data analysis will be performed using statistical correlation between the results of AAS and each indicator of the organization. Each individual analysis of this type represents the general results for an organization. The consolidation will take place from an analysis of quartiles to indicate the trend of results.

## **Final considerations**

The result of this research is not intended to be definitive, but rather intends to open the discussion about the validity and necessity of the existence of such methods. Other results can be considered depending on the population of companies obtained. One is to evaluate the behavior of the AAS in CMMI companies that have a measuring system mounted and can incorporate the result of this evaluation as indicator.

## **Discussion of Results**

These results should be evaluated very carefully, first by not yet defined the selection criteria on which companies may or may not participate. But the main focus of the discussions of the results is inherent in the nature of the improvement process itself.

First observation is to discard the idea that the best result of the AAS does not allow us to bring more process improvement and consequently an increase in ROI. This is false, in first plane, because both AAS which were chosen have a finite number of predefined questions that do not cover all aspects of agile are involved in a software organization

And also because the authors of this paper believe that the relationship between AAS and the ROI in the context of software process improvement has not a linear relationship. In fact, there may be changes in the process which are so expensive that your return will be negative, i.e., the AAS score increases but there is a decrease in the final ROI.

Thus, the best result is obtained at the vertex of the parabola, but which companies are left or right of the vertex, which are near or far significantly likely to impact the final result. Probably the outliers should be identified and the taken off the sample.

## **Implications for Research and Practice**

The implications of this work in the industry is able to provide organizations a quick way to assess their performance based on a quantitative indicator which is easy for be obtained.

Thus, even the goal of AAS is not being the rank of teams and individuals they could be used as guides for software process improvement within organizations.

In the research field the implication is to identify the usefulness of AAS as well as verification of alignment with SPI. The research result, whether positive or negative, open discussion about how it should be and what is the importance of a guide for improvements.

## **Limitations and Threats to Validity**

In addition to the points raised in the discussion of results, others threats to validity are raised from the selected method which is the case study.

The first limitation aimed at this aspect is the no possibilities of indicating relation of cause-effect since the lack of control over the environment only allow describing the problem and not explain it. But this same phenomenon occurs only in real life industry, and it occurs in different way for each particular company, there is no other means to present a study about issue.

A threat identified is to validity is the amount of companies involved in the study. A number of small companies cannot bring the desired strength to the study. Another point also refers to the number of iterations evaluated in each organization. A small number of iterations can also disable the data of a particular company.

Another threat comes from the validity of the statistical method used in this work. The lack of previous attempts prevents to determine which type of method is most appropriate for conduct the case study. So the proposition of using statistic correlations may not be the best choice.

A final threat to validity identified is the way data are collected from the ASA. In the case of past events, important details may be forgotten. Another threat to validity is that these tests are susceptible to mood swings of an individual who may be positively or negatively influenced by external factors.

## Future Work

If the correlation between AAS results and ROI results provides a strong correlation, both, linear and non-linear regression could be performed to identify the true relationship between the two variables. It is expected that the relationship shows itself as a non-linear. These regressions will be performed in each company individually, and they must be consolidated as well.

## Acknowledgments

Célio Santana is a doctoral student at the Center of Informatics of the Federal University of Pernambuco where he receives the funding from the Brazilian National Research Council (CNPq), process #141156/2010-4. The authors thank CNPq for partially funding the participation in the Agile@2011 Conference.

## References

- [1] M. Paulk, B. Curtis, M. Chrissis, and C. Weber, "Capability Maturity Model for Software (Version 1.1)", CMU/SEI-93-TR-024. Software Engineering Institute (SEI). February 1993.
- [2] F. G. Wilkie, D. McFall, and F. McCaffery, "An Evaluation of CMMI Process Areas for Small- to Medium-sized Software Development Organizations." In *Software Process Improvement and Practice*, Vol. 10, 2 (2), 2005, pp. 189-201.
- [3] M. C. Paulk, "Extreme Programming from a CMM Perspective." In *Software*, Vol. 18, 6 (6), Nov. 2011, pp. 19-26.
- [4] P. Kettunen and M. Laanti, "How to steer an embedded software project: tactics for selecting the software process model," *Information and Software Technology*, 2004.
- [5] C. Vriens, "Certifying for CMM Level 2 and ISO9001 with XP@Scrum," presented at Agile Development Conference, 2003.
- [6] O. Salo, "Enabling Software Process Improvement in Agile Software Development Teams and Organisations" PhD Thesis, Oulu, (2007).
- [7] M. Pikkarainen, and U. Passoja, "An Approach for Assessing Suitability of Agile Solutions: A Case Study." In: *The proceedings of the Sixth International Conference on Extreme Programming and Agile Processes in Software Engineering*. Sheffield University, UK. 2005, pp. 171-179.
- [8] B. Boehm and R. Turner, "Balancing Agility and Discipline," in *Balancing Agility and Discipline - A Guide for the Perplexed*: Addison Wesley, 2003.
- [9] S. W. Humphrey, "Discipline for Software Engineering." Addison Wesley Longman. 1995.
- [10] W. A. Florac, A. D. Carleton, and J. R. Barnard, "Statistical Process Control: Analyzing a Space Shuttle Onboard Software Process." *IEEE Software*, Vol. 17, 4 (4). July 2000, pp. 97-106.
- [11] D. F. Rico, "ROI of Software Process Improvement: Metrics for Project Managers and Software Engineers." J. Ross Publishing. Florida, U.S.A. 2004.

- [12] H. Krasner, "The Payoff for Software Process Improvement: What it is and How to Get it." In: Elements of Software Process Assessment & Improvement. El Emam, K.& Madhavji, N. H. (ed.). IEEE Computer Society, 1999, pp 113-130.
- [13] P. Lappo and C. T. A. Henry, "Assessing Agility," presented at Extreme programming and Agile Processes in Software Engineering, Germany, 2004.
- [14] B. Vodde, "Nokia Networks and Agile Development." In Proceedings of EuroMicro Conference. 2006.
- [15] J. Sutherland, "Money for nothing: And your change for free of Agile Contracts." In Proceedings of Agile Development Conference. 2008.
- [16] M. Cohn, "Succeeding with Agile: Software Development Using Scrum." Addison Wesley Longman. 2009.
- [17] M. Pikkarainen, F. Mc Caffery and I. Richardson "AHAA - Agile, Hybrid Assessment Method for Automotive, Safety Critical SMEs." International Conference on Software Engineering, Leipzig, Germany, 2008.
- [18] J. Paul, "Quantitative Agile Assessment Method." Master's Thesis, university of Turku. 2008.

## Does Agile Development Fit in the Actual Context of Software Quality?

Célio Santana, Cristine Gusmão, Hélio Bentzem, Alexandre Vasconcelos  
 Universidade Federal Rural de Pernambuco, Universidade Federal de Pernambuco, Universidade Federal de Pernambuco, Universidade Federal de Pernambuco, Unidade Pernambuco, Núcleo de Pernambuco, Centro de Pernambuco, Centro de Acadêmica de Garanhuns. Telesaúde (NUTES). Informática (Cin). Recife., Informática (Cin). Recife, Garanhuns. Brazil. Recife, Brazil. Brazil. Brazil.  
 Celio.santana@gmail.com, cristine.gusmao@nutes.ufpe.br, hfbpf@cin.ufpe.br, amlv@cin.ufpe.br

**Abstract** *Since 1968 the computer science tried to adopt elements of the traditional engineering in order to address the problems of quality and failures in the development of information systems. Many software quality references such as CMMI or six sigma are based in traditional engineering. Agile development reveals that it is hard to meeting the changing requirements right up to the level of product deployment. This paper presents how agile development needs could compromise its relationship with traditional quality software approaches.*

**Keywords** *Software Process Improvement; Agile Processes; Software Quality.*

### I. INTRODUCTION

In 1996 Barbara Kitchenham e Shari Pfleeger stated that software quality presents five views [1].

- É Transcendental View;
- É User View;
- É Manufacturing View;
- É Product View;
- É Value-Base View.

The existence of these views is one indicator of the complexity involving quality in software environment. The authors suggested these five visions based in concepts proposed by David A. Garvin [2] respected practitioners of quality in the industry. In fact this was not the first attempt of the software industry to use the concepts of the manufacturing industry when referring to quality. Long before this paper, in fact since 1968, when software engineering was born in NATO conference [3], software engineers have tried to emulate traditional engineering in order to address quality problems and Information Systems failures with varying degrees of success [4,5].

In 1975 was finished Goodenough [6] published that some the programs verified by the scientific community and formally stated as correct, presented errors in some outputs given inputs! It was established that there was not possible to verify the correctness of a program only through formal demonstration of the theorem. From this point there was the first turn in the concept of software quality since that formal demonstration of theorems was winding down while the quality control enthusiasts had create the culture of quality software product driven by the use of tests.

The work presented by Kitchenham & Pfleeger [1] was written in an important time for the software quality that was the mid-1990 when another turn happened. At the end of 80s

and beginning of 90s the emphasis was put on software process improvement (SPI) [7]. In this period of time emerged most of the quality models and approaches such as Quality Improvement Paradigm (QIP) [8], IDEALSM [9], ISO/IEC 15504 Part 7 [10], Capability Maturity Model (CMM) [11], ISO/IEC 12207 [12], ISO 9001-3 [13], ISO 9126 [14], Six Sigma for software development [15].

A new approach emerged at the end of 1990, being formalized in 2001 as Agile Methodologies. On a practical perspective, agile methodologies came out from a common discovery among practitioners who had slowly drifted away from the traditional software engineering [16]. This new perspective was soon evaluated in light of current concepts of software quality and of improvement and low control of the changes performed lead software development to more complex and not well-documented systems and to a fragmented software development process [16,17].

Shortly after, the initial concepts of quality have been formalized for agile methodologies. McBreen [18] defines agile quality assurance as the development of software that can respond to change, as the customer requires it to change. This implies that the frequent delivery of tested, working, and customer-approved software at the end of each iteration is an important aspect of agile quality assurance.

Huo et al. [19] developed a comparison technique whose aim was to provide a comparative analysis between quality in the waterfall development model (as a representative of the traditional camp) and quality in the agile group of methodologies. The results of the analysis showed that there is indeed quality assurance in agile development, but it is achieved in a different way from the traditional processes.

The limitations of Huo et al.'s tool however, are that the analysis:

- Singles out two main aspects of quality management namely quality assurance and verification and validation;
- Overlooks other vital techniques used in agile processes to achieve higher quality management.
- Agile quality assurance takes quality issues a step beyond the traditional software quality assurance approaches.

Ambler [20] considers agile quality to be a result of practices such as effective collaborative work, incremental development, and iterative development as implemented through techniques such as refactoring, test-driven development, modeling, and effective communication techniques.

Siakas & Siakas [21] presented one approach that the quality in Agile Development based in Human Factor. Information Systems quality requires knowledge of different organizational and national cultures, the methods and tools used, the ways they are used and, most importantly. The major issue identified in Agile and the Quality provided in Traditional Software Development is that Agility innovatively opens new horizons in the area of software quality assurance [7,20,21].

The motivation of this research is to investigate in the same perspective how quality must be addressed in an environment of rapid organizational level changes, people-centered and with low control of external events or agile environments. The objective is to determine how directions are made towards quality in agile organizations. What things are important to notice in environments where the control is low and changes are faster. This research is an ongoing MSc Thesis and this work is a research in Progress. After this introductory section, the section 2 presents the theoretical references about Traditional Software Quality Approaches, the section 3 shows the research structure, in section 4 final considerations are exposed.

## II. SOFTWARE QUALITY

Returning to the initial idea that the quality in software industry is based on the manufacturing industry, this point is clear when the traditional quality models and approaches are investigated. In follow subsections will be shown how each quality model is influenced by manufacture quality.

### CMMI

Nowadays, the most recognized software quality model is the Capability Maturity Model Integration [22] which is an updated version of CMM [11]. CMM presents many aspects of quality used in manufactured and consolidate by some respected quality

practitioners. For example, the CIM approach for using improving cycles comes from the Shewhart cycle (Also known as PDCA - Plan, Do, Check, Act) disseminated by Deming [23].

The adoption of the Product and Process Quality Assurance (PPQA) committee for address quality issues in company and the identification of Improve Opportunities and adopt quality projects comes from Juran [11,24]. The Five Levels of Maturity and capacity comes from Crosby concept [25] where the lowest level (level 1) the company does not know why exists problems quality and the highest level (level 5) the company knows why there is no problem of quality in its process.

## SIX SIGMA

Nowadays, The introduction of Six Sigma into the manufacturing arena in the early 1980s by Motorola was a step in revolutionizing the scope and use of quality systems in business today [15]. Shewhart introduced three sigma as a measurement of output variation in 1922, and stated that process intervention was needed when the output went beyond this limit. The three sigma concept is related to a process yield of 99.973 per cent and represented a defect rate of 2,600 per million, which was adequate for most manufacturing organizations until the early 1980s [15].

## ISO 9126

In 1987 Garvin suggested that there were eight dimensions to product quality which are:

1. Performance;
2. Features;
3. Reliability;
4. Durability;
5. Conformance;
6. Serviceability;
7. Aesthetics;
8. Perceived Quality.

The main idea of Garvin is that a product can present high quality in one dimension but can fail in other aspect. These dimensions are basis for the "Product Characteristics" found in ISO 9126 for software/.

## AGILE

Even agile software development addressing some of the 14 points of quality proposed by Deming [26] it is still limited in quality approach. Limitations were raised by other researches [7, 19, 21] because agile are and it is the lack of research which is target of this research.

## III. THE RESEARCH STRUCTURE

The research is based on an observational case study to evaluate how quality are addressed in companies which environments are "change-driven" and low control of the facts and these companies are using agile. Other information about the research is showed in the following subsections.

### A. Premises

The manufacturing system of production that inspired software engineers in the way of adopt quality in their industry presents two important characteristics: Controlled Environment: Shewhart proposed the Statistical Process Control and Deming proposed the Total Quality Management (TQM) what are approaches that aims to control the process more effectively and efficiently put a company in an excellent level of quality while maintaining

uncontrolled process, with variations allowed within a maximum range. The control charts and cause effect diagrams are artifacts commonly used in these approaches.

**Optimization Improvement:** In controlled environments the majority of improvements are towards optimization, since the company knows its process and it needs to be improved so that it produces faster, better, cheaper and with better quality while maintaining its level of quality within the accuracy expected. When adopting these models in software industry, it was accepted that the software industry presents the same characteristics of control and optimization. But it is not true because it is recognized that software development environments are complex adaptive systems where companies are low control and innovation is more valued than optimization [27].

This research will be taken as a premise that innovation companies must have a different way for achieve quality compared with innovation companies and these quality characteristics are the subject of the study. The hard task is to find innovation oriented software development since that majority of these companies are agile later adopt (not in vanguard) and are not change driven. It is common Internet companies being candidates to this case study.

## **B. Research Phases**

The first stage of research is to finish the theoretical background between optimization and innovation companies and how agile (innovation) are really different of traditional software development (optimization). The second stage of the research is the selection of companies where the case study will be conducted. From there, will be identified which ones are innovation oriented also identified the key organizational indicators of each company for future quantitative studies.

The next phase is the preparation of forms that will be used in the data collections generated in each organization. This form must be replicated to the companies because containing questions about quality in their environments. The last phase will be the creation of spreadsheets that will assist in analyzing the data received of each company individually, as well the consolidated of data.

## **C. Data Collection**

The forms provided to the companies will be collected even using web forms or on site forms. The monitoring of these companies will be longitudinal, and then the end of each cycle the companies must deliver the completed forms. Even now some internet companies are candidates to answer the survey.

## **D. Data Analysis**

Data analysis will be performed using a qualitative approach, the preference will be given to grounded theory if the number of companies is enough to do it, but probably will be changed. Each individual analysis of this type represents the general results for an organization. The consolidation will take place using thematic analysis.

## **IV. FINAL CONSIDERATIONS**

The result of this research is not intended to be definitive, but rather intends to open the discussion about the quality needs in innovative environments. Other results can be considered depending on the population of companies obtained. One is to evaluate quality aspects in innovation companies which use traditional approaches such as CMMI companies.

### **A. Discussion of Results**

These results should be evaluated very carefully, first by not yet defined the selection criteria on which companies may or may not participate. But the main focus of the discussions of the results is inherent in the aspects of quality itself. First observation is to find which characteristics are presented for innovation companies and just after that, find the aspects of quality that are presented in these companies worried with innovation (not optimization).

### **B. Implications for Research and Practice**

The implications of this work in the industry is enable innovation software development companies adopt agile processes conscious of the practices which can improve their quality. This contribution will be more valuable if aspects not addressed in traditional software development. In the research field the implication is to identify new quality perspectives in software development.

### **C. Limitations and Threats to Validity**

In addition to the points raised in the discussion of results, others threats to validity are raised from the selected method which is the case study. The first limitation aimed at this aspect is the no possibilities of indicating relation of cause-effect since the lack of control over the environment only allow describing the problem and not explain it. A threat identified is to validity is the amount of companies involved in the study. A number of small companies cannot bring the desired strength to the study. Another point also refers to the number of iterations evaluated in each organization. A small number of iterations can also disable the data of a particular company. A final threat to validity identified is the way data are collected (Surveys). In the case of past events, important details may be forgotten. Another threat to validity is that these tests are susceptible to mood swings of an individual who may be positively or negatively influenced by external factors.

## **REFERENCES**

- [1] B. Kitchenham & S. Pfleeger. "Software Quality: The elusive target". IEEE Software, vol 13, pp 12-21. January, 1996.
- [2] D. Garvin, D. "Managing quality". New York: Macmillan. 1988.
- [3] P. Naur, & B. Randell, B. "Software engineering: Report on a Conference Sponsored by the NATO Science Committee. Paper presented at the 1st NATO Software Engineering Conference, Garmisch, Germany. October. 1968.
- [4] E. Georgiadou. "Software Process and Product Improvement: A Historical Perspective. Cybernetics, and Systems Analysis, 11(4), pp 25-142. 2003.
- [5] I. Sommerville. "Software engineering (9th ed.)". Pearson Education. 2010.
- [6] J. B. Goodenough & S. L. Gerhart. "Toward a Theory of Test Data Selection". IEEE Transaction on Software Engineering, vol SE-1, N. 2, pp. 156-172. June, 1975.
- [7] I. Stamelos & P. Sifatsos. "Agile Software Development Quality Assurance. Information Science Reference, Hershey. 2007.
- [8] V. R. Basili. "Software Development: A Paradigm for the Future". In: The proceedings of the COMPSAC'89, Orlando, Florida. pp. 471-485. 1989.
- [9] B. McFeeley. IDEALSM: A Users Guide for Software Process Improvement. CMU/SEI-96-HB-001. 1996.
- [10] H. Van Loon. "Process Assessment and ISO/IEC 15504: a reference book". Springer-Verlag New York. 2007.
- [11] M. Paulk, B. Curtis, M. Chrissis, C. Weber. Capability Maturity Model for Software (Version 1.1). 1993.
- [12] ISO/IEC 12207:2000 - Information technology software process life cycle, (2000)
- [13] ISO 9001:2000 - Quality management systems - Requirements, (2000).

- [14] International Organization for Standardisation. (ISO) ISO/IEC: 9126 Information technology- Software Product Evaluation-Quality characteristics and guidelines for their use 1991. [verified 10 Oct 2011] <http://www.cse.dcu.ie/essiscope/sm2/9126ref.html>
- [15] Motorola University Press, Utilizing the Six Steps to Six Sigma, Schaumburg, IL. 1993.
- [16] B. Boehm & R. Turner, Balancing agility and discipline: A guide for the perplexed 1<sup>st</sup> ed. Addison Wesley, pp. 165-194. 2004.
- [17] B. Boehm & R. Turner, Using risk to balance agile and plan-driven methods. Computer, 36(6), 57-64. 2003.
- [18] P. McBreen, Quality assurance and testing in agile projects. 2003. McBreen Consulting. Retrieved October 12, 2011, from <http://www.mcbreen.ab.ca/talks/CAMUG.pdf>
- [19] M. Huo, J. Verner, L. Zhu, & M. A. Babar. Software quality and agile methods. Proceedings of the 28th Annual International Computer Software and Applications Conference (COMPSAC04). IEEE Computer. 2004.
- [20] S. Ambler. Quality in an agile world. Software Quality Professional 7(4), 34-40. 2005.
- [21] E. Siakas & K. Siakas. The Human Factor Deployment for Improved Agile Quality, in Markku Tukianen, Richard Messnards, Risto Nevalaninen, Sonja Koining (eds), European Software Process Improvement and Innovation (EuroSPI 2006), International Proceedings Series 6, 11-13 October, Joensuu, Finland, Publisher: University of Joensuu, ISBN: 952-458-864-1, pp. 4.11- 23. 2006.
- [22] Software Engineering Institute. CMMI for Development (CMMI-DEV), Version 1.2, Technical Report CMU/SEI-2006-TR-008. Pittsburgh, PA. 2006.
- [23] M. R. Yilmaz, & Sangit, C. Deming and the quality of software development. In Business Horizon. 6 Ed. Vol 40, pp 51-58. 1997.
- [24] J. M. Juran, Juran on Leadership for Quality. Free Press. 1989.
- [25] P. Crosby. Quality is Free. New York: McGraw-Hill. 1979.
- [26] W. E. Deming. Out of the Crisis. 10<sup>th</sup> Printing ed. 1990.
- [27] R. Darnal & J. M. Preston. Project Management from Simple to Complex. 2010.

## Utilização de Práticas Genéricas do CMMI para apoiar a utilização de Metodologias Ágeis.

Célio Santana, Cristine Gusmão, Ana Rouiller, Alexandre Vasconcelos

Universidade de Pernambuco, Departamento de Sistemas e Computação, Rua do Benfica,  
455, 50720-001 Recife, Pernambuco

Universidade Federal Rural de Pernambuco, Departamento de Estatística e Informática,  
Av. Dois Irmãos. S/N, 52171-900 Recife, Pernambuco

Universidade Federal de Pernambuco, Centro de Informática, Av. Prof. Luiz Freire. S/N,  
50740-540 Recife, Pernambuco

celio.santana@ati.pe.gov.br, cmgg@dsc.upe.br, ana@deinfo.ufrpe.br, amlv@cin.ufpe.br.

**Resumo.** Este trabalho apresenta como o modelo tradicional de qualidade de software Capability Maturity Model Integration (CMMI), através de suas práticas genéricas, pode apoiar as Metodologias Ágeis apesar da aparência antagônica entre as duas abordagens. Este apoio, não só se refere à aderência das metodologias ao modelo em si, mas também, considerar como aspectos ágeis podem ser aprimorados com as melhores práticas propostas pelo CMMI. Este trabalho é resultado da avaliação deste problema em uma organização que pretende alcançar o CMMI nível 2.

**Abstract.** This paper shows how the traditional software quality model Capability Maturity Model Integration (CMMI), through its generic practices, could support Agile Methodologies despite the opposite appearance between those approaches. This kind of support does not concern only in adherence between that methodologies and the model, but, also how agile aspects could be improved by using the best practices proposed by CMMI. This paper is a result of the evaluation about this issue in a company that intends to reach CMMI level 2.

**Palavras Chave:** Metodologias Ágeis, Capability Maturity Model Integration, CMMI, Modelos de Qualidade de Software.

### 1 Introdução

No início da década de 1990 o modelo de qualidade *Capability Maturity Model for Software* (SW-CMM) [1] se tornou o modelo tradicional mais reconhecido no mercado. Em 2001 foi criado o Manifesto Ágil [2] que propunha valores e práticas que foram denominadas Metodologias Ágeis e era baseado em resposta rápida a mudanças e pouca documentação seguindo um caminho antagônico pregado a uma visão orientada a planejamento oferecida pelos modelos tradicionais.

Mesmo com o antagonismo aparente entre as duas abordagens, em 2001, surgiu um trabalho [3] que propunha a adequação ao SW-CMM de uma abordagem ágil denominada *Extreme Programming* (XP) [4]. A partir daí surgiram novos trabalhos adequando o CMMI a outras abordagens ágeis, contudo nenhum deles considerava o uso das Metas e Práticas Genéricas tão importantes no CMMI [6] bem como os valores e Princípios do Manifesto Ágil.

Este trabalho apresenta a experiência de uma organização que utiliza Metodologias Ágeis e está em processo de adequação ao CMMI nível 2 em utilizar as práticas genéricas para apoiar a implementação das metodologias ágeis.

### 2 Metodologias Ágeis

O termo Metodologias Ágeis foi formalizado por 17 pessoas resultando em um acordo que possuía quatro níveis [5]. O primeiro nível considera a necessidade da existência de métodos construídos para responder a mudanças durante o projeto de software. O segundo nível do acordo foi a publicação do Manifesto Ágil [2] que é composto por quatro valores centrais de todas as metodologias:

- indivíduos e iterações mais do que processos e ferramentas;
- Software funcionando mais do que documentação extensa;
- Colaboração do cliente mais do que negociação de contratos;
- Responder a mudanças mais do que seguir um plano.

Embora haja valor nos termos a direita, são valorizados mais os termos a esquerda. O terceiro nível é composto por um conjunto de doze princípios. O quarto e último nível do acordo é o nível no qual cada abordagem ágil deveria definir a si mesma.

### 3 CMMI

O CMMI [6] é um modelo de maturidade para melhoria de processos de software. Ele consiste das melhores práticas que tem como alvo atividades de desenvolvimento e manutenção que cobre todo o ciclo de vida do produto. Nele existem duas abordagens: a representação contínua e a representação estagiada. Na representação contínua cada área de processo pode atingir seis níveis de capacidade (0 ao 5) enquanto na representação estagiada, existem cinco níveis de maturidade (1 a 5).

Para alcançar o nível dois de maturidade, uma organização deve estar com todas as áreas de processo relacionadas a este nível de maturidade no mínimo nível dois de capacidade. Para entender a importância das práticas genéricas no CMMI devemos observar a obtenção de níveis de capacidade de cada área de processo [6]:

**Nível 0** ó O nível zero significa que a organização não consegue realizar todas as práticas específicas daquela área de processo.

**Nível 1** ó O nível um, indica que a empresa consegue atender a todas as práticas específicas daquela área, porém não atende as práticas genéricas de nível dois.

**Nível 2 ao 5** ó Nesses níveis é necessário que a organização atenda ao nível imediatamente inferior àquele desejado bem como as práticas genéricas daquele nível para a área de processo, mas não atende à todas as práticas genéricas do nível superior.

### 4 Utilização das Práticas Genéricas

A organização onde foi conduzido o estudo não será identificada por motivos de confidencialidade. Está situada em Recife (Brasil) e possui 30 funcionários alocados para o desenvolvimento de software separado em quatro times de Scrum [8] e deseja obter o nível 2 de maturidade do CMMI em um ano.

Houve a preocupação da adequação do Scrum às práticas específicas das áreas de processo: Planejamento de Projetos, Monitoramento de Projetos, Gerenciamento de Requisitos. Mas, a preocupação não era apenas com a adequação, mas, com o apoio que o CMMI provia ao Scrum. Esse suporte é fornecido pelas as práticas genéricas do CMMI.

#### 4.1 Práticas Genéricas da Área de Processo Planejamento de Projetos

A Prática Genérica 2.2 informa que se deve planejar o processo de planejamento de projetos. O Scrum define as responsabilidades de cada papel, quais são as reuniões de planejamento existentes, a obrigação de cada papel dentro destas reuniões e como cada reunião deve ser conduzida. Contudo, o próprio time pode definir regras próprias, nesta organização, é utilizada a técnica Planning Poker [8] para estimativas. Esta forma de planejamento é apoiada para projetos Scrum observando esta Prática Genérica que além de suportar as práticas do Scrum apóia a utilização de regras próprias que podem ser diferentes de uma equipe para outra.

A Prática Genérica 2.3 informa que se deve prover todos os recursos para que ocorra o planejamento de projetos como foi definido. São providenciados a equipes e quadros para confecção de áreas de acompanhamento de projetos, post-its, e material de papelaria para a

organização do espaço de trabalho. Uma sala foi alocada para cada tipo de reunião e nelas são encontradas informações sobre o projeto.

A Prática Genérica 2.4 informa que se deve atribuir as responsabilidades para o planejamento de projetos. O próprio Scrum já informa as responsabilidades de cada papel dentro das reuniões de planejamento. Porém são incorporadas aqui as responsabilidades para as regras estabelecidas pelo time.

A Prática Genérica 2.5 informa que deve haver algum treinamento para o planejamento de projetos. Todos os quatro times foram treinados em cursos internos providos pela empresa e pela participação de membros de uma determinada equipe em reuniões de planejamento de outras equipes mais experientes.

A Prática Genérica 2.7 informa que todos os interessados relevantes devem ser identificados para o planejamento de projetos. O Scrum identifica todos os envolvidos na atividade de planejamento de projetos e atribui as suas responsabilidades.

A Prática Genérica 2.8 informa que o processo de planejamento de projetos deve ser monitorado e controlado, já a prática 2.9 indica que o processo deve ser avaliado objetivamente. Essas duas práticas estão ligadas ao papel do *ScrumMaster*. É obrigação de ele avaliar se o processo está sendo seguido de acordo com as regras da organização e as regras colocadas pelo time. Durante as reuniões diárias, ele deve verificar se algum membro do time não está seguindo o processo, seja voluntária ou involuntariamente, e caso haja desvios ele deve tomar alguma atitude para que as coisas funcionem como acordado.

As Práticas Genéricas 2.4 (atribuir responsabilidades), 2.5 (treinar pessoas), 2.7 (identificar e envolver interessados relevantes), 2.8 (monitorar e controlar o processo) e 2.9 (avaliar o processo de forma objetiva) serão implementadas da mesma forma que ocorreram aqui para as demais áreas de processo com exceção das áreas Garantia da Qualidade (2.8 e 2.9 apenas) e Gerência de Configuração (todas as cinco práticas).

#### 4.2 Práticas Genéricas da Área de Processo Monitoramento e Controle de Projetos

A Prática Genérica 2.2 informa que se deve planejar o processo de Monitoramento e Controle de projetos. O Scrum define que a reunião diária deve ser usada para esse fim. Caso haja impedimentos o *ScrumMaster* deve removê-los para que a equipe possa seguir da forma mais produtiva possível. A Prática Genérica 2.3 informa que se deve prover todos os recursos necessários para que o monitoramento e controle de projetos ocorra como foi definido. Além dos recursos providenciados no planejamento, uma nova sala é alocada para as reuniões em pé.

#### 4.3 Práticas Genéricas da Área de Processo Garantia da Qualidade do Produto e Processo

A Prática Genérica 2.2 informa que se deve planejar o processo de Garantia da Qualidade do Produto e do Processo. O Scrum possui quatro práticas que estão relacionadas a esta área de processo: A primeira é a definição do *ScrumMaster* como responsável por garantir que o Scrum esteja sendo seguido como acordado (Práticas Genéricas 2.8). A segunda é a reunião de Retrospectivas, aonde o time pode propor mudanças nas regras (no processo) para realizar o seu trabalho. A terceira é a indicação de que uma tarefa pronta é aquela testada e aprovada para todos os casos de testes, e neste caso, a definição de pronto verifica se o código já está integrado à linha base. E a última é a reunião de Revisão da Sprint com a aprovação das funcionalidades pelo *Product Owner*.

A Prática Genérica 2.3 informa que se deve prover todos os recursos necessários para que a garantia da qualidade do produto e processo ocorra como foi definida. Aqui foi alocada uma nova sala para que a equipe fizesse a reunião de retrospectivas. Foi também alocada uma sala para apresentação da nova versão do sistema ao *Product Owner*. Foi adquirido pela organização ferramentas para automatização dos testes bem como um servidor de integração contínua para auxiliar na garantia da qualidade do produto final.

Vale ressaltar que as Práticas Genéricas 2.8 (monitorar e controlar o processo) e 2.9 (avaliar o processo de forma objetiva) não são previstas pelo Scrum. Quer dizer, o Scrum não

específica como monitorar e controlar as atividades do *ScrumMaster* verificando se o mesmo realiza suas tarefas de forma aderente ao processo, sendo esta obrigação da organização prover tal controle. Neste caso, a organização coloca os outros três *ScrumMasters* para verificar como foi o trabalho do outro restante.

#### 4.4 Práticas Genéricas da Área de Processo Gerência de Requisitos

A Prática Genérica 2.2 informa que se deve planejar o processo de Gerência de Requisitos. O Scrum define que o *Product Backlog* deve ser o local onde os requisitos do *Product Owner* devem ser armazenados, e que o *Sprint Backlog* deve conter todos os requisitos que estão sendo concebidos naquele Sprint. Define também quem prioriza esses itens e quando eles devem mudar de um para outro, sugere critérios de aceitação para definição de pronto e indica que estes requisitos devem ser homologados pelo *Product Owner*. Essa prática genérica apóia todo o processo de definição de produto sugerido pelo Scrum.

A Prática Genérica 2.3 informa que se deve prover todos os recursos necessários para que a Gerência de Requisitos. O espaço para o backlog com quadro branco e post-its são providenciados pela organização.

#### 4.5 Práticas Genéricas da Área de Processo Gerência de Configuração

Para esta área de processo não há nada específico no Scrum, porém com a utilização de prática de XP integração contínua algumas observações devem ser realizadas. A Prática Genérica 2.3 informa que se deve prover todos os recursos necessários para que a gerência de configuração ocorra como foi definida. Aqui foram adquiridas ferramentas de repositório e versionamento de código fonte, ferramenta para automação de testes, um servidor de integração contínua e um gerador automático *debuilds*. Estas ferramentas estão integradas para manter a integridade do código.

A Prática Genérica 2.4 informa que se deve atribuir as responsabilidades para a Gerência de Configuração. A própria equipe definiu regras de quem, quando e como as ferramentas devem ser utilizadas pelo time. Quem não segue estas regras acaba quebrando o *build*. Assim o *ScrumMaster* verifica com certa frequência quem quebrou o *build* para tomar ações corretivas.

A Prática Genérica 2.5 informa que deve haver algum treinamento para a Gerência de Configuração. Todos os quatro times foram treinados em cursos internos providos pela empresa para o uso do sistema de gestão de configuração. A Prática Genérica 2.8 informa que o processo de deve ser monitorado e controlado, já a prática 2.9 indica que o processo deve ser avaliado objetivamente. É obrigação do *ScrumMaster* verificar se houve quebra do *build*. Se isso ocorrer, ações corretivas devem ser tomadas.

## 5 Conclusão

Este trabalho mostrou que mesmo se mostrando historicamente antagônicos, o *Capability Maturity Model Integration* (CMMI) e as Metodologias Ágeis podem ser utilizados em conjunto em um mesmo processo, sendo o CMMI um instrumento de apoio para a institucionalização das Metodologias Ágeis. Esse suporte se dá por partes das práticas genéricas do modelo.

### 5.1 Dificuldades encontradas

As principais dificuldades encontradas neste contexto foram:

- O nível 2 do CMMI possui poucas práticas genéricas para apoio das Metodologias Ágeis. A organização acredita que outras práticas genéricas dos níveis mais altos podem auxiliar a integração entre as abordagens;
- A utilização das outras práticas genéricas que não são atendidas pelo Scrum podem engessar o processo se não forem adequadamente tratadas;

□ Quando o time decide mudar o processo, é necessário que haja um projetopiloto para avaliá-la objetivamente. A forma como o Scrum propõe as retrospectivas podem conflitar com a área de processo PPQA, pois, o projetopiloto é um *Sprint* do próprio projeto e a avaliação é baseada no sentimento do time que não é objetiva.

## 5.2 Trabalhos Futuros

Alguns trabalhos futuros são propostos:

- Medir indicadores da empresa e observar se há melhoria nesta forma conjunta para o desenvolvimento de software;
- Tentar avaliar se em níveis mais altos do CMMI se as práticas genéricas suportam as Metodologias Ágeis de forma mais completa;

## Referências

1. Paulk, M. C., Curtis, B., Chrissis, M. B. Capability Maturity Model V 1.1. IEEE Software 10(4), pp. 19--27 (1993)
2. Agile Manifesto, <http://www.agilemanifesto.org/> - Último acesso em 04/11/2008.
3. Paulk, M. C. Extreme Programming from a CMM perspective, IEEE, vol. 18, (2001)
4. Beck, K.: Extreme Programming Explained: Embrace Change. Addison Wesley, Reading (1999)
5. Koch, A. S.: Agile Software Development - Evaluating the Methods for Your Organization. Artech House, Boston (2005)
6. Chrissis, M. B., Konrad, M., Shrum, S.: CMMI® For Development Version 1.2: Guidelines for Process Integration and Product Improvement Second Edition. Addison Wesley. (2007)
7. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice Hall (2001)
8. Grenning, J.: Planning Poker, disponível em: [www.objectmentor.com/resources/articles/PlanningPoker.zip](http://www.objectmentor.com/resources/articles/PlanningPoker.zip) .(2002)

## Anexo A o Nokia Test.

Responda o questionário abaixo considerando a última iteração.

NOKIA TEST	
Nome do Projeto	Desenvolvedor
Parte 1 - Desenvolvimento Iterativo	
Questão 1 : As iterações acontecem a cada 4 semanas ou menos?	
RESPOSTAS	PONTOS
Sem iterações	0
Iterações > 6 semanas	1
Iteração com tamanho variável < 6 semanas	2
Iteração com tamanho Fixo = 6 semanas	3
Iteração com tamanho Fixo = 5 semanas	4
Iteração com tamanho Fixo <= 4 semanas	10
Questão 2 : Os recursos de Software devem ser testados e está funcionando no final de cada iteração?	
RESPOSTAS	PONTOS
Não há garantia da qualidade	0
Testes Unitários	1
Testes funcionalidades	5
Testes funcionalidades logo que concluídas	7
Software aprovado em testes de aceitação	8
Software é implantado	10
Questão 3 : A iteração deve começar antes da especificação dos requisitos está completa?	
RESPOSTAS	PONTOS
Sem requisitos	0
Documentos de requisitos grandes	1
Histórias de usuário pobres	4
Bons requisitos	5
Boas histórias de usuário	7
Apenas o suficiente, apenas na hora de especificar	8
Boas histórias de usuários vinculados às especificações que forem necessárias	10

Parte 2 - Scrum	
Questão 4 : Você sabe quem é o product owner?	
RESPOSTAS	PONTOS
Não há Product Owner	0
Product Owner não entende Scrum	1
Product Owner que interrompe a equipe	2
Product Owner não se envolve com a equipe	2
Product Owner com product backlog	5
Product Owner com as datas do release baseado na velocidade do time	8
Product Owner que motiva a equipe	10
Questão 5 : Existe um product backlog priorizado business value?	

RESPOSTAS	PONTOS
Sem Product Backlog	0
Vários Product Backlogs	1
Simple Product Backlog	3
Product Backlog priorizado por ROI	5
Product Owner tem plano de release com base em Product Backlog	7
Product Owner pode medir ROI baseado no rendimento real, custo por story point , ou outras métricas	10
<b>Questão 6 : O product backlog tem estimativas criadas pela equipe?</b>	
RESPOSTAS	PONTOS
Product Backlog não é estimado	0
Estimativas não são criadas pela equipe	1
Estimativas não são criadas pelo planning poker	5
Estimativas criadas pelo planning poker pela equipe	8
Erros de estimativas < 10%	10
<b>Questão 7 : A equipe gera gráficos burndown e conhece sua velocidade?</b>	
RESPOSTAS	PONTOS
Não há gráfico burndown	0
Gráfico burndown não é atualizado pela equipe	1
Gráfico burndown em horas/dias não é contabilizado em trabalho em andamento	2
Gráfico burndown só queima quando a tarefa é concluída	4
Gráfico burndown só queima quando uma story é concluída	5
Adicione 3 pontos se a equipe conhece velocidade	
Adicione 2 pontos se o Product Owner planeja release baseado na velocidade conhecida	
<b>Questão 8 : Não existem gerentes de projeto (ou alguém) para interromper o trabalho da equipe?</b>	
RESPOSTAS	PONTOS
Gerente ou líder do projeto interrompe a equipe	0
Product Owner interrompe a equipe	1
Gerentes, Líderes de projeto ou Líderes de equipe atribuem tarefas	3
Tem Líder de Projeto e papéis Scrum	5
Ninguém interrompe a equipe, somente papéis Scrum	10